Introduction to Deep Inference

Lecture notes for ESSLLI'19

August 5–16, 2019, University of Latvia

Andrea Aler Tubella

Umeå University
901 87 Umeå— Sweden

https://aalertubella.com/

Lutz Straßburger

INRIA Saclay – Île-de-France
École Polytechnique, Laboratoire d'Informatique (LIX)
Rue de Saclay — 91128 Palaiseau Cedex — France

http://www.lix.polytechnique.fr/~lutz/

Contents

0 What is this?

These are the notes for a 5-lecture-course given during the first week of ESSLLI'19, held from August 5 to 16, 2010, at The University of Latvia (Riga). The URL of the school is

`http://esslli2019.folli.info`

The course will give a basic introduction to deep inference, which is a design principle for proof formalisms in which inference rules can be applied at any depth inside the proof. In this course, we will provide a clear understanding of the intuitions behind deep inference, together with a rigorous account of the properties that deep inference proof systems enjoy, especially in regards to normalisation. Properties that particularly stand out are atomicity, locality, and regularity. Normalisation procedures specific to deep inference allow for new notions of normal forms for proofs, as well as for a general modular cut-elimination theory. Furthermore, the ability to track every occurrence of an atom throughout a proof allows for the definition of geometric invariants with which it is possible to normalise proofs without looking at their logical connectives or logical rules, obtaining a purely geometric normalisation procedure.

This course is intended to be introductory. That means no prior knowledge of proof theory is required. However, the student should be familiar with the basics of propositional logic.

1 What are proof formalisms, and why do we need them?

Already in ancient Greece people tried to formalize the notion of a logical argument. For example, the rule of *modus ponens*, in modern notation written as

$$\mathsf{mp}\,\frac{A \qquad A \rightarrow B}{B} \tag{1}$$

goes back at least to Aristoteles. The figure in (1) says that if you know that A is true and you also know that A implies B, then you can conclude B.

In the early 20th century David Hilbert had the idea to formalize mathematics. He wanted to prove its consistency in order to avoid paradoxes (like *Russel's paradox*). Although this plan failed, due to Gödel's Incompleteness Theorem, Hilbert's work had huge impact on the development of modern proof theory. He introduced the first *formal deductive system* consisting of *axioms* and *inference rules.*

1.1 Hilbert systems

Figure 1 shows a so-called *Hilbert system* (also called *Frege systems* or *Hilbert-Frege-systems* or *Hilbert-Ackermann-systems*) for classical propositional logic. The system in Figure 1, that we call here H, contains ten axioms and one rule: modus ponens.

More precisely, we should speak of ten *axiom schemes* and one *rule scheme.* Each axiom scheme represents infinitely many axioms. For example

$$(a \wedge c) \rightarrow ((a \vee (b \wedge \neg c)) \rightarrow (a \wedge c))$$

is an instance of the axiom scheme

$$A \rightarrow (B \rightarrow A)$$

Notation 1.1.1. Throughout this lecture notes, we use lower case latin letters a, b, c, ..., for propositional variables, and capital latin letters A, B, C, ..., for formula variables. As usual, the symbol $\wedge$ stands for conjunction (*and*), $\vee$ stands for disjunction (*or*), and $\rightarrow$ stands for implication. Furthermore, to ease the reading of long formulas, we use different types of brackets for the different connectives. We use $(\ldots)$ for $\wedge$, $(\ldots)$ for $\vee$, and $(\ldots)$ for $\rightarrow$. This is pure redundancy and has no deep meaning.

A *proof* in a Hilbert system is a sequence of formulas $A_0, A_1, A_2, \ldots, A_n$, where for each $0 \leq i \leq n$, the formula A_i is either an axiom, or it follows from A_j and A_k via modus ponens, where $j, k < i$. The formula A_n is called the *conclusion* of the proof.

The main results on Hilbert systems are *soundness* and *completeness*:

Theorem 1.1.2 (Soundness)**.** *If there is a proof in* H *with conclusion A, then A is a tautology.*

Theorem 1.1.3 (Completeness)**.** *If the formula A is a tautology, then there is a proof in* H *with conclusion A.*

$$
\begin{array}{ll}
A \to (B \to A) & (A \wedge B) \to A \\
(A \to (B \to C)) \to (A \to B) \to A \to C & (A \wedge B) \to B \\
A \to (A \vee B) & A \to (B \to (A \wedge B)) \\
B \to (A \vee B) & \mathsf{f} \to A \\
(A \to C) \to (B \to C) \to ((A \vee B) \to C) & \neg\neg A \to A
\end{array}
$$

$$
\mathsf{mp}\,\frac{A \quad A \to B}{B}
$$

Figure 1: The Hilbert system H

The main achievement of Hilbert systems is that they made proofs into first-class mathematical objects that could be manipulated by mathematical means, and about which theorems could be stated and proved. Proof theory as a mathematical field was born.

However, Hilbert systems are not easy to use to actually prove stuff. Just as an exercise, try to prove Pierce's law in a Hilbert system.

Exercise 1.1.4. Prove Pierce's law $((A \to B) \to A) \to A$ in the Hilbert system shown in Figure 1.

1.2 Natural deduction

If you tried this exercise, you might have noticed that proving stuff in a Hilbert system can be quite tedious. For this reason, Gerhard Gentzen introduced the notion of *natural deduction* (in German *Natürliches Schließen*) which resembles more closely the way mathematicians reason in mathematical proofs.

Figure 2 shows his system NK. Let us more closely inspect some of the rules:

- $\wedge$I: This rule is called $\wedge$-introduction, because it introduces an $\wedge$ in the conclusion. It says: if there is a proof of A and a proof of B, then we can form a proof of $A \wedge B$ which has as assumptions the union of the assumptions of the proofs of A and B.

- $\to$I: This rule is called $\to$-introduction, because introduces an $\to$. It says that if we can prove B under the assumption A, then we can prove $A \to B$ without that assumption. The notation $\cancel{A}$ simply says that A had been removed from the list of assumptions.

- $\to$E: This rule is called $\to$-elimination, because it eliminates an $\to$. It is exactly the same as modus ponens.

Exercise 1.2.1. Find similar explanations for the other rules.

$$
\wedge\mathrm{I}\,\dfrac{\overset{\Pi_1}{A} \qquad \overset{\Pi_2}{B}}{A \wedge B}
\qquad
\wedge\mathrm{E_R}\,\dfrac{\overset{\Pi_1}{A \wedge B}}{A}
\qquad
\wedge\mathrm{E_L}\,\dfrac{\overset{\Pi_1}{A \wedge B}}{B}
$$

$$
\to\mathrm{I}\,\dfrac{\overset{\overset{\cancel{A}}{\Pi_1}}{B}}{A \to B}
\qquad
\to\mathrm{E}\,\dfrac{\overset{\Pi_1}{A \to B} \qquad \overset{\Pi_2}{A}}{B}
$$

$$
\vee\mathrm{I_R}\,\dfrac{\overset{\Pi_1}{A}}{A \vee B}
\qquad
\vee\mathrm{I_L}\,\dfrac{\overset{\Pi_1}{B}}{A \vee B}
\qquad
\vee\mathrm{E}\,\dfrac{\overset{\Pi_1}{A \vee B} \qquad \overset{\overset{\cancel{A}}{\Pi_2}}{C} \qquad \overset{\overset{\cancel{B}}{\Pi_3}}{C}}{C}
$$

$$
\neg\mathrm{I}\,\dfrac{\overset{\overset{\cancel{A}}{\Pi_1}}{\mathsf{f}}}{\neg A}
\qquad
\neg\mathrm{E}\,\dfrac{\overset{\Pi_1}{\neg A} \qquad \overset{\Pi_2}{A}}{\mathsf{f}}
\qquad
\mathsf{f}\mathrm{E}\,\dfrac{\overset{\Pi_1}{\mathsf{f}}}{C}
\qquad
\neg\neg\mathrm{E}\,\dfrac{\overset{\Pi_1}{\neg\,\neg\, A}}{A}
$$

Figure 2: The natural deduction system NK

Example 1.2.2. Let us now see an example proof:

$$
\to\mathrm{I}\,\dfrac{\vee\mathrm{E}\,\dfrac{A \vee \cancel{(B \wedge C)} \qquad \wedge\mathrm{I}\,\dfrac{\vee\mathrm{I_R}\,\dfrac{\cancel{A}}{A \vee B} \quad \vee\mathrm{I_R}\,\dfrac{\cancel{A}}{A \vee C}}{(A \vee B) \wedge (A \vee C)} \qquad \wedge\mathrm{I}\,\dfrac{\vee\mathrm{I_R}\,\dfrac{\wedge\mathrm{E_R}\,\dfrac{\cancel{B \wedge C}}{B}}{A \vee B} \quad \vee\mathrm{I_R}\,\dfrac{\wedge\mathrm{E_L}\,\dfrac{\cancel{B \wedge C}}{C}}{A \vee C}}{(A \vee B) \wedge (A \vee C)}}{(A \vee B) \wedge (A \vee C)}}{(A \vee (B \wedge C)) \to ((A \vee B) \wedge (A \vee C))} \tag{2}
$$

Informally, we can read this proof as follows: We want to prove

$$(A \vee (B \wedge C)) \to ((A \vee B) \wedge (A \vee C))$$

We assume $A \vee (B \wedge C)$. There are two cases: We have A or we have $B \wedge C$. In the first case we can conclude $A \vee B$ as well as $A \vee C$, and therefore also $(A \vee B) \wedge (A \vee C)$. In the second case we can conclude B and C, and therefore also $A \vee B$ as well as $A \vee C$, from which we get $(A \vee B) \wedge (A \vee C)$. We have therefore shown $(A \vee B) \wedge (A \vee C)$ from the assumption $A \vee (B \wedge C)$, and we can conclude $(A \vee (B \wedge C)) \to ((A \vee B) \wedge (A \vee C))$.

As for Hilbert systems, we have soundness and completeness for NK.

Theorem 1.2.3 (Soundness)**.** *If there is a proof in* NK *with conclusion* A*, then* A *is a tautology.*

Theorem 1.2.4 (Completeness)**.** *If the formula A is a tautology, then there is a proof in* NK *with conclusion A.*

Exercise 1.2.5. Use the system NK (shown in Figure 2) for proving the axioms of the system H (shown in Figure 1).

Exercise 1.2.6. If you did Exercises 1.1.4 and 1.2.5 then you can immediately produce a proof of Pierce's law in NK. How? Can you find a simpler proof of Pierce's law $((A \to B) \to A) \to A$ in NK?

1.3 Sequent calculus

In order to reason about derivations in natural deduction, Gentzen also introduced the *sequent calculus.* Figure 3 shows his system LK. While Hilbert systems have many axioms and few rules, sequent systems have few axioms and many rules. Gentzen's original system (Figure 3) is a variant of what is nowadays called a *two-sided system*, where a *sequent*

$$A_1, \ldots, A_n \vdash B_1, \ldots, B_m \tag{3}$$

consists of two lists of formulas, and should be read as: The conjunction of the A_i entails the disjunction of the B_j. As formula:

$$(A_1 \wedge \cdots \wedge A_n) \to (B_1 \vee \cdots \vee B_m)$$

Lists of formulas are usually denoted by capital greek letters, like Γ, Δ, Λ, ...

As for Hilbert systems an natural deduction, we have soundness and completeness for LK.

Theorem 1.3.1 (Soundness)**.** *If there is a proof in* LK *with conclusion* $\vdash A$ *, then A is a tautology.*

Theorem 1.3.2 (Completeness)**.** *If the formula A is a tautology, then there is a proof in* LK *with conclusion* $\vdash A$ *.*

Example 1.3.3. To give an example how the rules work, we prove here the same formula as in Example 1.2.2:

$$
{\scriptstyle \to\mathrm{R}}\,\dfrac{
{\scriptstyle \vee\mathrm{L}}\,\dfrac{
{\scriptstyle \wedge\mathrm{R}}\,\dfrac{
{\scriptstyle \vee\mathrm{R}_1}\,\dfrac{{\scriptstyle \mathsf{id}}\,\dfrac{}{A \vdash A}}{A \vdash A \vee B}
\quad
{\scriptstyle \vee\mathrm{R}_1}\,\dfrac{{\scriptstyle \mathsf{id}}\,\dfrac{}{A \vdash A}}{A \vdash A \vee C}
}{A \vdash (A \vee B) \wedge (A \vee C)}
\quad
{\scriptstyle \wedge\mathrm{R}}\,\dfrac{
{\scriptstyle \vee\mathrm{R}_2}\,\dfrac{{\scriptstyle \wedge\mathrm{L}_1}\,\dfrac{{\scriptstyle \mathsf{id}}\,\dfrac{}{B \vdash B}}{B \wedge C \vdash B}}{B \wedge C \vdash A \vee B}
\quad
{\scriptstyle \vee\mathrm{R}_2}\,\dfrac{{\scriptstyle \wedge\mathrm{L}_2}\,\dfrac{{\scriptstyle \mathsf{id}}\,\dfrac{}{C \vdash C}}{B \wedge C \vdash C}}{B \wedge C \vdash A \vee C}
}{B \wedge C \vdash (A \vee B) \wedge (A \vee C)}
}{A \vee (B \wedge C) \vdash (A \vee B) \wedge (A \vee C)}
}{\vdash (A \vee (B \wedge C)) \to ((A \vee B) \wedge (A \vee C))}
\tag{4}
$$

Exercise 1.3.4. Prove the axioms of the system H (shown in Figure 1) with the sequent calculus LK (shown in Figure 3). Pay special attention to the cut-rule; which axioms of H can you prove without it, and for which axioms do you need the cut-rule?

$$\mathsf{id}\,\frac{}{A \vdash A}$$

$$\mathsf{weak}\mathrm{L}\,\frac{\Gamma \vdash \Theta}{A, \Gamma \vdash \Theta} \qquad \mathsf{weak}\mathrm{R}\,\frac{\Gamma \vdash \Theta}{\Gamma \vdash \Theta, A}$$

$$\mathsf{cont}\mathrm{L}\,\frac{A, A, \Gamma \vdash \Theta}{A, \Gamma \vdash \Theta} \qquad \mathsf{cont}\mathrm{R}\,\frac{\Gamma \vdash \Theta, A, A}{\Gamma \vdash \Theta, A}$$

$$\mathsf{exch}\mathrm{L}\,\frac{\Delta, B, A, \Gamma \vdash \Theta}{\Delta, A, B, \Gamma \vdash \Theta} \qquad \mathsf{exch}\mathrm{R}\,\frac{\Gamma \vdash \Theta, B, A, \Lambda}{\Gamma \vdash \Theta, A, B, \Lambda}$$

$$\wedge\mathrm{L}_1\,\frac{A, \Gamma \vdash \Theta}{A \wedge B, \Gamma \vdash \Theta} \quad \wedge\mathrm{L}_2\,\frac{B, \Gamma \vdash \Theta}{A \wedge B, \Gamma \vdash \Theta} \qquad \wedge\mathrm{R}\,\frac{\Gamma \vdash \Theta, A \qquad \Gamma \vdash \Theta, B}{\Gamma \vdash \Theta, A \wedge B}$$

$$\vee\mathrm{L}\,\frac{A, \Gamma \vdash \Theta \qquad B, \Gamma \vdash \Theta}{A \vee B, \Gamma \vdash \Theta} \qquad \vee\mathrm{R}_1\,\frac{\Gamma \vdash \Theta, A}{\Gamma \vdash \Theta, A \vee B} \quad \vee\mathrm{R}_2\,\frac{\Gamma \vdash \Theta, B}{\Gamma \vdash \Theta, A \vee B}$$

$$\to\mathrm{L}\,\frac{\Gamma \vdash \Theta, A \qquad B, \Delta \vdash \Lambda}{A \to B, \Gamma, \Delta \vdash \Theta, \Lambda} \qquad \to\mathrm{R}\,\frac{A, \Gamma \vdash \Theta, B}{\Gamma \vdash \Theta, A \to B}$$

$$\neg\mathrm{L}\,\frac{\Gamma \vdash \Theta, A}{\neg A, \Gamma \vdash \Theta} \qquad \neg\mathrm{R}\,\frac{A, \Gamma \vdash \Theta}{\Gamma \vdash \Theta, \neg A}$$

$$\mathsf{cut}\,\frac{\Gamma \vdash \Theta, A \qquad A, \Delta \vdash \Lambda}{\Gamma, \Delta \vdash \Theta, \Lambda}$$

Figure 3: Gentzen's sequent calculus LK

Observe that in natural deduction there are *introduction rules* and *elimination rules*, whereas in the sequent calculus there are only introduction rules: *introduction on the left* and *introduction on the right*. The rules for contraction (contL and contR), weakening (weakL and weakR), and exchange (exchL and exchR) are called *structural rules* because they modify only the "structure" of the sequent. The rules for $\wedge$, $\vee$, $\to$, and $\neg$ are called *logical rules*. A special role is played by the id rule and by the cut rule, which, in a certain sense can considered duals of each other.

The rule id is the axiom. It says that A implies A. An interesting observation is that in the sequent calculus the identity axiom can be reduced to an atomic version

$$\mathsf{atomic\ id}\,\frac{}{a \vdash a} \tag{5}$$

Proposition 1.3.5. *The rule* id *is derivable in the system* $\{\mathsf{atomic\ id}\} \cup \mathsf{LK} \setminus \{\mathsf{id}\}$.

Proof. Suppose we have an instance of id:

$$\mathsf{id}\,\frac{}{A \vdash A}$$

We proceed by induction on the size of A to construct a derivation that uses only the atomic version of id.

- If $A = B \wedge C$, then we can replace

$$\mathsf{id}\,\frac{}{B \wedge C \vdash B \wedge C} \qquad \text{by} \qquad \wedge\mathrm{R}\,\frac{\wedge\mathrm{L}_1\,\dfrac{\mathsf{id}\,\dfrac{}{B \vdash B}}{B \wedge C \vdash B} \qquad \wedge\mathrm{L}_2\,\dfrac{\mathsf{id}\,\dfrac{}{C \vdash C}}{B \wedge C \vdash C}}{B \wedge C \vdash B \wedge C} \tag{6}$$

  and proceed by induction hypothese.

The other cases are similar (see Execise 1.3.6). □

Exercise 1.3.6. Complete the proof of Proposition 1.3.5 (i.e., show the cases that are omitted).

The cut rule expesses the transitivity of the logical consequence relation: if from B we can conclude A, and from A we can conclude C, then from B we can conclude C directly. One can say that the cut rule allows to use "lemmas" in a proof. The main and most surprising result for the sequent calculus LK is that if there is a proof in LK, then the same conclusion can be proved in LK *without* using the cut rule. This is nowadays called *cut elimination*.

Theorem 1.3.7. *If a sequent* $\Gamma \vdash \Theta$ *is provable in* LK*, then it is also provable in* $\mathsf{LK} \setminus \{\mathsf{cut}\}$.

(Sketch). We do not show the complete proof here, but we will discuss the basic proof idea and the biggest problem to overcome. Let us consider a topmost instance of cut in a proof in LK. I.e., we have a situation:

$$\mathsf{cut}\,\frac{\dfrac{\Pi_1}{\Gamma \vdash \Theta, A} \qquad \dfrac{\Pi_2}{A, \Delta \vdash \Lambda}}{\Gamma, \Delta \vdash \Theta, \Lambda} \tag{7}$$

where Π_1 and Π_2 do not contain instances of cut, i.e., are *cut-free*. Usually, the cut elimination proceeds by an induction measure that includes the size of cut-formula A and the sizes of one or both of Π_1 and Π_2 and maybe another value incorporating arrangements of the instances of contraction in the proof. Then the cut elimination proof usually proceeds by a case analysis on the bottommost rule instances in Π_1 and Π_2. The simplest case is when one of these two rule instances does not involve the cut-formula A. Then we can perform a simple rule permutation:

$$\mathsf{cut}\,\dfrac{\wedge\mathrm{L}_1\,\dfrac{\begin{array}{c}\Pi'_1\\ B,\Gamma'\vdash\Theta,A\end{array}}{B\wedge C,\Gamma'\vdash\Theta,A}\qquad\begin{array}{c}\Pi_2\\ A,\Delta\vdash\Lambda\end{array}}{B\wedge C,\Gamma',\Delta\vdash\Theta,\Lambda}\qquad\rightsquigarrow\qquad\wedge\mathrm{L}_1\,\dfrac{\mathsf{cut}\,\dfrac{\begin{array}{c}\Pi'_1\\ B,\Gamma'\vdash\Theta,A\end{array}\qquad\begin{array}{c}\Pi_2\\ A,\Delta\vdash\Lambda\end{array}}{B,\Gamma',\Delta\vdash\Theta,\Lambda}}{B\wedge C,\Gamma',\Delta\vdash\Theta,\Lambda}$$

and proceed by induction hypothesis. If both rules decompose the cut formula, we can *reduce* the cut. For example, assume $A = B \wedge C$, and we have a situation as shown on the left below:

$$\mathsf{cut}\,\dfrac{\wedge\mathrm{R}\,\dfrac{\begin{array}{c}\Pi'_1\\ \Gamma\vdash\Theta,B\end{array}\qquad\begin{array}{c}\Pi''_1\\ \Gamma\vdash\Theta,C\end{array}}{\Gamma\vdash\Theta,B\wedge C}\qquad\wedge\mathrm{R}_1\,\dfrac{\begin{array}{c}\Pi'_2\\ B,\Delta\vdash\Lambda\end{array}}{B\wedge C,\Delta\vdash\Lambda}}{\Gamma,\Delta\vdash\Theta,\Lambda}\qquad\rightsquigarrow\qquad\mathsf{cut}\,\dfrac{\begin{array}{c}\Pi'_1\\ \Gamma\vdash\Theta,B\end{array}\qquad\begin{array}{c}\Pi'_2\\ B,\Delta\vdash\Lambda\end{array}}{\Gamma,\Delta\vdash\Theta,\Lambda}$$

This can be replaced by a single cut on the subformula B, as shown on the right above. Note that (i) the size of the cut formula is reduced (which is the reason that we can apply the induction hypothesis) and (ii) the subproof Π''_1 has been deleted. It can also happen, that a subproof is duplicated, as in the situation on the left below, where the cut formula is subject to a contraction:

$$\mathsf{cut}\,\dfrac{\mathsf{contR}\,\dfrac{\begin{array}{c}\Pi'_1\\ \Gamma\vdash\Theta,A,A\end{array}}{\Gamma\vdash\Theta,A}\qquad\begin{array}{c}\Pi_2\\ A,\Delta\vdash\Lambda\end{array}}{\Gamma,\Delta\vdash\Theta,\Lambda}\qquad\rightsquigarrow\qquad\mathsf{contL}\,\dfrac{\mathsf{contR}\,\dfrac{\mathsf{cut}\,\dfrac{\mathsf{cut}\,\dfrac{\begin{array}{c}\Pi'_1\\ \Gamma\vdash\Theta,A,A\end{array}\qquad\begin{array}{c}\Pi_2\\ A,\Delta\vdash\Lambda\end{array}}{\Gamma,\Delta\vdash\Theta,\Lambda,A}\qquad\begin{array}{c}\Pi_2\\ A,\Delta\vdash\Lambda\end{array}}{\Gamma,\Delta,\Delta\vdash\Theta,\Lambda,\Lambda}}{\Gamma,\Delta,\Delta\vdash\Theta,\Lambda}}{\Gamma,\Delta\vdash\Theta,\Lambda}\tag{8}$$

When we try to permut this instance of the contraction rule below the cut, we have to replace the cut-instance by two instances of cut with the same cut formula A, and many instances of contraction below the two cuts, as shown on the right above. If the bottommost rule in Π_2 is a contL-instance on A, then it is easy to see that this never terminates and the proof only gets bigger and bigger. To solve this problem, Gentzen intoduced the *Mischung*, now also known as *Mix* or *multi-cut*:[1]

$$\mathsf{mcut}\,\dfrac{\Gamma\vdash\Theta,A,\ldots,A\qquad A,\ldots,A,\Delta\vdash\Lambda}{\Gamma,\Delta\vdash\Theta,\Lambda}$$

where and arbitrary number of A are introduced on both premises. Then the situation in (8) can be resolved as follows:

[1]The name *Mix* has many different meanings in different proof theory communities. For this reason, we stick here to *multi-cut*.

$$
\mathsf{mcut}\,\dfrac{\mathsf{contR}\,\dfrac{\overset{\Pi_1'}{\Gamma \vdash \Theta, A, \ldots, A, A}}{\Gamma \vdash \Theta, A, \ldots, A} \qquad \overset{\Pi_2}{A, \ldots, A, \Delta \vdash \Lambda}}{\Gamma, \Delta \vdash \Theta, \Lambda}
\quad \rightsquigarrow \quad
\mathsf{mcut}\,\dfrac{\overset{\Pi_1'}{\Gamma \vdash \Theta, A, \ldots, A, A} \qquad \overset{\Pi_2}{A, \ldots, A, \Delta \vdash \Lambda}}{\Gamma, \Delta \vdash \Theta, \Lambda}
$$

Where we can proceed by induction hypothesis because the height of the proof branch on the left is reduced. □

Exercise 1.3.8. Complete the proof for Theorem 1.3.7. In particular, list all cases and find an induction measure that is reduced in each case. Then you can conclude that the reduction procedure terminates.

Some consequences of cut elimination (in propositional logic and in first order predicate logic) are the *subformula property* and the *consistency* of the system.

The subformula property says that every formula that occurs somewhere in the proof is a subformula of the conclusion. It is easy to see that only the cut rule violates this property in LK.

Consistency says that there is no formula A such that we can prove both A and $\neg A$. This can be proved as follows: By way of contradiction assume we have such a formula. By using the cut rule, we can derive the empty sequent $\vdash$. By cut elimination there is a cut-free proof of the empty sequent $\vdash$. But by the subformula property this is impossible.

One-sided sequent calculus

If the logic has DeMorgan duality (like classical logic), we only need to consider formulas in negation normal form, i.e., negation is pushed to the atoms via the DeMorgan laws:

$$\neg(A \wedge B) = \neg A \vee \neg B \qquad \neg(A \vee B) = \neg A \wedge \neg B \qquad \neg\neg A = A \tag{9}$$

and implication is eliminated by using

$$A \to B = \neg A \vee B \tag{10}$$

Then we need to consider only *one-sided sequents*:

$$\vdash B_1, \ldots, B_m \tag{11}$$

In such a system, negation is often denoted by $\overline{(\cdot)}$, i.e., we write $\bar{A}$ instead of $\neg A$.

The translation of a two-sided sequent (3) into a one-sided sequent is simply $\vdash \bar{A}_1, \ldots, \bar{A}_n, B_1, \ldots, B_m$

The practical advantage is that we can halve the number of rules. Figure 4 shows the one-sided version of LK.

There are many different sequent systems for classical logic; a second one is shown in Figure 5. One-sided systems are also called *Gentzen-Schütte* systems.

Exercise 1.3.9. Translate the axioms of the Hilbert system H into negation normal form, and prove them using the rules in Figure 4.

Exercise 1.3.10. Show that the two systems in Figures 4 and 5 are equivalent, i.e., prove the same sequents.

$$\mathsf{id}\ \frac{}{\vdash \bar{A}, A}$$

$$\mathsf{weak}\ \frac{\vdash \Gamma}{\vdash \Gamma, A} \qquad \mathsf{cont}\ \frac{\vdash \Gamma, A, A}{\vdash \Gamma, A} \qquad \mathsf{exch}\ \frac{\vdash \Delta, B, A, \Gamma}{\vdash \Delta, A, B, \Gamma}$$

$$\wedge\ \frac{\vdash \Gamma, A \quad \vdash \Gamma, B}{\vdash \Gamma, A \wedge B} \qquad \vee_1\ \frac{\vdash \Gamma, A}{\vdash \Gamma, A \vee B} \qquad \vee_2\ \frac{\vdash \Gamma, B}{\vdash \Gamma, A \vee B}$$

$$\mathsf{cut}\ \frac{\vdash \Gamma, A \quad \vdash \bar{A}, \Delta}{\vdash \Gamma, \Delta}$$

Figure 4: One-sided version of LK

$$\mathsf{weak}\ \frac{\vdash \Gamma}{\vdash \Gamma, A} \qquad \mathsf{cont}\ \frac{\vdash \Gamma, A, A}{\vdash \Gamma, A} \qquad \mathsf{exch}\ \frac{\vdash \Delta, B, A, \Gamma}{\vdash \Delta, A, B, \Gamma}$$

$$\mathsf{id}\ \frac{}{\vdash a, \bar{a}} \qquad \wedge\ \frac{\vdash \Gamma, A \quad \vdash \Delta, B}{\vdash \Gamma, A \wedge B, \Delta} \qquad \vee\ \frac{\vdash \Gamma, A, B}{\vdash \Gamma, A \vee B} \qquad \mathsf{cut}\ \frac{\vdash \Gamma, A \quad \vdash \bar{A}, \Delta}{\vdash \Gamma, \Delta}$$

Figure 5: Another one-sided sequent calculus for classical logic

$$\mathsf{id}\ \frac{}{\vdash A^{\perp}, A} \qquad \mathsf{exch}\ \frac{\vdash \Delta, B, A, \Gamma}{\vdash \Delta, A, B, \Gamma} \qquad \mathsf{cut}\ \frac{\vdash \Gamma, A \quad \vdash A^{\perp}, \Delta}{\vdash \Gamma, \Delta}$$

$$\oplus_1\ \frac{\vdash \Gamma, A}{\vdash \Gamma, A \oplus B} \qquad \oplus_2\ \frac{\vdash \Gamma, B}{\vdash \Gamma, A \oplus B} \qquad \&\ \frac{\vdash \Gamma, A \quad \vdash \Gamma, B}{\vdash \Gamma, A \mathbin{\&} B}$$

Figure 6: Sequent system ALL^{-} for additive linear logic without units

$$\mathsf{id}\ \frac{}{\vdash A^{\perp}, A} \qquad \mathsf{exch}\ \frac{\vdash \Delta, B, A, \Gamma}{\vdash \Delta, A, B, \Gamma} \qquad \mathsf{cut}\ \frac{\vdash \Gamma, A \quad \vdash A^{\perp}, \Delta}{\vdash \Gamma, \Delta}$$

$$\parr\ \frac{\vdash \Gamma, A, B}{\vdash \Gamma, A \oplus B} \qquad \otimes\ \frac{\vdash \Gamma, A \quad \vdash B, \Delta}{\vdash \Gamma, A \otimes B, \Delta}$$

Figure 7: Sequent system MLL^{-} for multiplicative linear logic without units

Linear logic

If you compare again the two systems in Figures 4 and 5, you will observe that the only rules that are different are the ones for $\wedge$ and $\vee$. The variants in Figure 4 are called *additive* and the variants in Figure 5 are called multiplicative. If the rules of contraction and weakening are present, then the two variants are equivalent (as you have observed in Exercise 1.3.10).

But what happens if we remove contraction and weakening? Then we get different logics and different symbols for the connectives are used: $\&$ (*with*) and $\oplus$ (*plus*) for the additive conjunction and disjunction, and $\otimes$ (*tensor*) and $\parr$ (*par*) for the multiplicative conjunction and disjunction. Figures 6 and 7 show *additive* and *multiplicative linear logic (without units)*. Note that in linear logic, negation is denoted differently.

Exercise 1.3.11. What is the maximal/minimal number of formulas that can occur in a provable sequent in ALL^-? What about MLL^-?

Exercise 1.3.12. Prove cut elimination for ALL^- and MLL^-.

Exercise 1.3.13. We can combine the two systems into *multiplicative additive linear logic (MALL) without units.* Write down the rules for that system.

In classical logic, the units t (*truth*) and f (*falsum*) can be recovered via the formula $a_0 \vee \bar{a}_0$ and $a_0 \wedge \bar{a}_0$, respectively, for some fresh propositional variable a_0. However, in linear logic this is not possible, and this leads to four different units $\top$ (*top*), 1 (*one*), $\perp$ (*bottom*), and 0 (*zero*), defined via the rules:

$$\top \frac{}{\vdash \Gamma, \top} \qquad 1 \frac{}{\vdash 1} \qquad \perp \frac{\vdash \Gamma}{\vdash \Gamma, \perp} \qquad \text{no rule for } 0$$

Exercise 1.3.14. Which unit belongs to which connective? Why?

In multiplicative additive linear logic, no duplication or deletion is possible because there is no contraction and weakening. In order to recover these in a controlled way, linear logic has two modalities ? (called *why not*) and ! (called *of course* or *bang*), subject to the following inference rules:

$$! \frac{\vdash A, ?B_1, \ldots, ?B_n}{\vdash !A, ?B_1, \ldots, ?B_n} \qquad ?\mathsf{c} \frac{\vdash \Gamma, ?A, ?A}{\vdash \Gamma, ?A} \qquad ?\mathsf{w} \frac{\vdash \Gamma}{\vdash \Gamma, ?A} \qquad ?\mathsf{d} \frac{\vdash \Gamma, A}{\vdash \Gamma, ?A}$$

Finally, Figure 8 shows the sequent system for full propositional linear logic.

Exercise 1.3.15. Show that Proposition 1.3.5, i.e., that the atomic identiy rule

$$\text{atomic id} \frac{}{\vdash a^\perp, a}$$

can replace the rule id, also holds for the system LL in Figure 8. In other words, prove that id is derivable in $\{\text{atomic id}\} \cup \mathsf{LL} \setminus \{\mathsf{id}\}$.

Exercise 1.3.16. Prove cut elimination for LL.

$$
\mathsf{id}\,\frac{}{\vdash A^{\perp}, A}
\qquad
\mathsf{cut}\,\frac{\vdash \Gamma, A \qquad \vdash A^{\perp}, \Delta}{\vdash \Gamma, \Delta}
\qquad
\mathsf{exch}\,\frac{\vdash \Delta, B, A, \Gamma}{\vdash \Delta, A, B, \Gamma}
$$

$$
1\,\frac{}{\vdash 1}
\qquad
\otimes\,\frac{\vdash \Gamma, A \qquad \vdash B, \Delta}{\vdash \Gamma, A \otimes B, \Delta}
\qquad
\invamp\,\frac{\vdash \Gamma, A, B}{\vdash \Gamma, A \oplus B}
\qquad
\perp\,\frac{\vdash \Gamma}{\vdash \Gamma, \perp}
$$

$$
\top\,\frac{}{\vdash \Gamma, \top}
\qquad
\&\,\frac{\vdash \Gamma, A \qquad \vdash \Gamma, B}{\vdash \Gamma, A \mathbin{\&} B}
\qquad
\oplus_1\,\frac{\vdash \Gamma, A}{\vdash \Gamma, A \oplus B}
\qquad
\oplus_2\,\frac{\vdash \Gamma, B}{\vdash \Gamma, A \oplus B}
$$

$$
?\mathsf{d}\,\frac{\vdash \Gamma, A}{\vdash \Gamma, ?A}
\qquad
!\,\frac{\vdash A, ?B_1, \ldots, ?B_n}{\vdash !A, ?B_1, \ldots, ?B_n}
\qquad
?\mathsf{c}\,\frac{\vdash \Gamma, ?A, ?A}{\vdash \Gamma, ?A}
\qquad
?\mathsf{w}\,\frac{\vdash \Gamma}{\vdash \Gamma, ?A}
$$

Figure 8: System LL for full propositional linear logic

1.4 Calculus of structures

All proof formalisms that we have seen so far have one common feature: The proof progresses by manipulating the outermost connectives of the formula trees. In natural deduction and the sequent calculus it is only the root connective (or *main connective*) that is removed or introduced in an inference rule. This is the most important property that makes cut elimination work. In the following, we will call this kind of formalism *shallow inference* formalisms.

This brings us directly to the actual topic of this course: the *deep inference* formalism, that abandons the importance of the main connective. The first such formalism is the *calculus of structures*[2], which breaks with the tradition of the main connective and allows rewriting of formulas deep inside any context. It derives its name from the fact that there is no distinction between *sequents* and *formulas*, but there is a unique syntactic *structure* which can be seen as an equivalence class of formulas modulo associativity and commutativity and unit equations that are sometimes imposed on sequents (in the previous section the comma is associative and concatenation with the empty sequent does not change a sequent).

Figure 10 shows the equation that we use here to generate these equivalence classes, and Figure 9 shows system SKSg. However, one can avoid such an equational theory by incorporating the equations into the rules, as it is done in the system shown in Figure 11. That system has another property: all rules are *local.*

We have seen in Proposition 1.3.5 that in the sequent calculus, the identity axiom can be reduced to an atomic form. The same can be done for the corresponding rule in SKSg. However, by duality, we can do the same for the cut rule, which is not possible in the sequent calculus. Furthermore, if we add the rules

$$
\mathsf{nm}{\downarrow}\,\frac{S\{\mathsf{f}\}}{S\{\mathsf{f} \wedge \mathsf{f}\}}
\qquad
\mathsf{m}\,\frac{S\{(A \wedge B) \vee (C \wedge D)\}}{S\{(A \vee C) \wedge (B \vee D)\}}
\qquad
\mathsf{nm}{\uparrow}\,\frac{S\{\mathsf{t} \vee \mathsf{t}\}}{S\{\mathsf{t}\}}
\tag{12}
$$

[2]The original motivation for the calculus of structures was to overcome some restrictions of the sequent calculus which could not express a certain logic with a self-dual non-commutative connective. We will discuss this later in the course.

$$\mathsf{i}\!\downarrow \frac{S\{\mathsf{t}\}}{S\{A \vee \bar{A}\}} \qquad \mathsf{i}\!\uparrow \frac{S\{A \wedge \bar{A}\}}{S\{\mathsf{f}\}}$$

$$\mathsf{s} \frac{S\{(A \vee B) \wedge C\}}{S\{A \vee (B \wedge C)\}}$$

$$\mathsf{w}\!\downarrow \frac{S\{\mathsf{f}\}}{S\{A\}} \qquad \mathsf{w}\!\uparrow \frac{S\{A\}}{S\{\mathsf{t}\}}$$

$$\mathsf{c}\!\downarrow \frac{S\{A \vee A\}}{S\{A\}} \qquad \mathsf{c}\!\uparrow \frac{S\{A\}}{S\{A \wedge A\}}$$

Figure 9: The deep inference system SKSg for classical logic

$$\begin{array}{lll} A \wedge (B \wedge C) = (A \wedge B) \wedge C & A \wedge B = B \wedge A & A \wedge \mathsf{t} = A \\ A \vee (B \vee C) = (A \vee B) \vee C & A \vee B = B \vee A & A \vee \mathsf{f} = A \end{array}$$

Figure 10: Equational theory for SKSg

we can do the same with contraction and weakening, which is also impossible in the sequent calculus.[3]

Proposition 1.4.1. *The rules* $\mathsf{i}\!\downarrow$, $\mathsf{i}\!\uparrow$, $\mathsf{c}\!\downarrow$, $\mathsf{c}\!\uparrow$, $\mathsf{w}\!\downarrow$, *and* $\mathsf{w}\!\uparrow$ *are derivable in* SKS.

Proof. As in the proof of Proposition 1.3.5, we proceed by induction on the size of the principal formula of the rule.

- If $A = B \wedge C$, then we can do the following replacements:

$$\mathsf{i}\!\downarrow \frac{S\{\mathsf{t}\}}{S\{(B \wedge C) \vee \bar{C} \vee \bar{B}\}} \quad \to \quad \mathsf{s} \frac{\mathsf{i}\!\downarrow \dfrac{\mathsf{i}\!\downarrow \dfrac{S\{\mathsf{t}\}}{S\{B \vee \bar{B}\}}}{S\{(B \wedge (C \vee \bar{C})) \vee \bar{B}\}}}{S\{(B \wedge C) \vee \bar{C} \vee \bar{B}\}} \tag{13}$$

$$\mathsf{w}\!\downarrow \frac{S\{\mathsf{f}\}}{S\{B \wedge C\}} \quad \to \quad \mathsf{w}\!\downarrow \frac{\mathsf{w}\!\downarrow \dfrac{\mathsf{nm}\!\downarrow \dfrac{S\{\mathsf{f}\}}{S\{\mathsf{f} \wedge \mathsf{f}\}}}{S\{\mathsf{f} \wedge C\}}}{S\{B \wedge C\}} \tag{14}$$

$$\mathsf{c}\!\downarrow \frac{S\{(B \wedge C) \vee (B \wedge C)\}}{S\{B \wedge C\}} \quad \to \quad \mathsf{c}\!\downarrow \frac{\mathsf{c}\!\downarrow \dfrac{\mathsf{m} \dfrac{S\{(B \wedge C) \vee (B \wedge C)\}}{S\{(B \vee B) \wedge (C \vee C)\}}}{S\{(B \vee B) \wedge C\}}}{S\{B \wedge C\}} \tag{15}$$

[3] We will come back to this in the next section.

$$\mathsf{ai}\!\downarrow \frac{S\{\mathsf{t}\}}{S\{a \vee \bar{a}\}} \qquad \mathsf{ai}\!\uparrow \frac{S\{a \wedge \bar{a}\}}{S\{\mathsf{f}\}}$$

$$\mathsf{s} \frac{S\{A \wedge (B \vee C)\}}{S\{(A \wedge B) \vee C\}}$$

$$\mathsf{aw}\!\downarrow \frac{S\{\mathsf{f}\}}{S\{a\}} \qquad \mathsf{ac}\!\downarrow \frac{S\{a \vee a\}}{S\{a\}} \qquad \mathsf{ac}\!\uparrow \frac{S\{a\}}{S\{a \wedge a\}} \qquad \mathsf{aw}\!\uparrow \frac{S\{a\}}{S\{\mathsf{t}\}}$$

$$\mathsf{nm}\!\downarrow \frac{S\{\mathsf{f}\}}{S\{\mathsf{f} \wedge \mathsf{f}\}} \qquad \mathsf{m} \frac{S\{(A \wedge B) \vee (C \wedge D)\}}{S\{(A \vee C) \wedge (B \vee D)\}} \qquad \mathsf{nm}\!\uparrow \frac{S\{\mathsf{t} \vee \mathsf{t}\}}{S\{\mathsf{t}\}}$$

$$\alpha\!\downarrow \frac{S\{A \vee (B \vee C)\}}{S\{(A \vee B) \vee C\}} \qquad \sigma\!\downarrow \frac{S\{A \vee B\}}{S\{B \vee A\}} \qquad \sigma\!\uparrow \frac{S\{A \wedge B\}}{S\{B \wedge A\}} \qquad \alpha\!\uparrow \frac{S\{A \wedge (B \wedge C)\}}{S\{(A \wedge B) \wedge C\}}$$

$$\mathsf{f}\!\downarrow \frac{S\{A\}}{S\{A \vee \mathsf{f}\}} \qquad \mathsf{t}\!\downarrow \frac{S\{A\}}{S\{A \wedge \mathsf{t}\}} \qquad \mathsf{t}\!\uparrow \frac{S\{\mathsf{f} \vee A\}}{S\{A\}} \qquad \mathsf{f}\!\uparrow \frac{S\{\mathsf{t} \wedge A\}}{S\{A\}}$$

Figure 11: System SKS

In each case we can proceed by induction hypothesis. For the rules $\mathsf{i}\!\uparrow$, $\mathsf{c}\!\uparrow$, and $\mathsf{w}\!\uparrow$ the situation is similar.

- We leave the cases $A = B \vee C$, $A = \mathsf{t}$, $A = \mathsf{f}$, and $A = a$ as an exercise.

□

Exercise 1.4.2. Complete the proof of Proposition 1.4.1.

Proposition 1.4.3. *The rules* $\mathsf{nm}\!\downarrow$, $\mathsf{nm}\!\uparrow$, *and* m *are derivable in* SKSg.

Proof. The rules $\mathsf{nm}\!\downarrow$ and $\mathsf{nm}\!\uparrow$ are instances of $\mathsf{w}\!\downarrow$ and $\mathsf{w}\!\uparrow$, respectively. The rule m can be derived using $\mathsf{w}\!\downarrow$ and $\mathsf{c}\!\downarrow$ (see Exercise 1.4.4). □

Exercise 1.4.4. Show how medial can be derived using $\mathsf{w}\!\downarrow$ and $\mathsf{c}\!\downarrow$. Can you also derive medial using $\mathsf{w}\!\uparrow$ and $\mathsf{c}\!\uparrow$?

Exercise 1.4.5. Conclude that if there is a derivation from A to B in SKSg then there is one in SKS, and vice versa.

We use the following notation

$$\begin{array}{c} A \\ \mathscr{S}\Big\Vert\Pi \\ B \end{array} \qquad \text{and} \qquad \begin{array}{c} \mathscr{S}\Big\Vert\Pi' \\ B \end{array}$$

for denoting a derivation Π in system $\mathscr{S}$ from premise A to conclusion B, and a proof Π' of conclusion B in system $\mathscr{S}$, respectively, where a proof is a derivation with premise t.

Theorem 1.4.6 (Soundness and Completeness). *The formula $A \to B$ is a tautology if and only if there is a derivation*

$$\begin{array}{c} A \\ \mathsf{SKS}\,\Big\Vert\,\Pi \\ B \end{array}$$

Exercise 1.4.7. Prove the axioms of the Hilbert system H using SKSg.

The two systems in the calculus of structures that we presented so far have an interesting property. All inference rules come in pairs:

$$\rho\,\frac{S\{A\}}{S\{B\}} \qquad \text{and} \qquad \bar{\rho}\,\frac{S\{\bar{B}\}}{S\{\bar{A}\}} \tag{16}$$

where $\bar{\rho}$ is the *dual* of ρ, and is obtained from ρ by negating and exchanging premise and conclusion. For example, $\mathsf{c}{\downarrow}$ is the dual of $\mathsf{c}{\uparrow}$, and $\mathsf{i}{\uparrow}$ is the dual of $\mathsf{i}{\downarrow}$. The rules s and m are *self-dual.*

If the rules $\mathsf{i}{\downarrow}$, $\mathsf{i}{\uparrow}$, and s a derivable in a system $\mathscr{S}$, then $\mathscr{S}$ can derive for each rule also its dual:

Proposition 1.4.8. *Let ρ and $\bar{\rho}$ be a pair of dual rules. Then $\bar{\rho}$ is derivable in the system $\{\rho, \mathsf{i}{\downarrow}, \mathsf{i}{\uparrow}, \mathsf{s}\}$.*

Proof. The rule $\bar{\rho}$ can be derived in the following way:

$$\bar{\rho}\,\frac{S\{\bar{B}\}}{S\{\bar{A}\}} \quad \leadsto \quad \mathsf{i}{\uparrow}\,\frac{\mathsf{s}\,\frac{\rho\,\frac{\mathsf{i}{\downarrow}\,\frac{S\{\bar{B}\}}{S\{\bar{B} \wedge (a \vee \bar{A})\}}}{S\{\bar{B} \wedge (B \vee \bar{A})\}}}{S\{(\bar{B} \wedge B) \vee \bar{A}\}}}{S\{\bar{A}\}} \tag{17}$$

□

In a well-defined system in the calculus of structures, *cut elimination* means not only the admissibility of the cut-rule $\mathsf{i}{\uparrow}$, but the admissibility of the whole *up fragment*, i.e., all rules with an $\uparrow$ in the name. We will discuss cut elimination in deep inference several times in this couse.

1.5 Notes

As the name says, Hilbert systems have been introduced by David Hilbert [Hil22, HA28]. Gödel's Incompleteness Theorem has been published in [Göd31]. Natural Deduction and the sequent calculus have been introduced by Gerhard Gentzen in [Gen35a, Gen35b], where he also presented cut elimination. There is a similar result for natural deduction, called *normalization*, which has first been described by Dag Prawitz [Pra65]. A standard textbook on proof theory, treating these issues in more detail is [TS00]. Linear logic has been introduced by Jean-Yves Girard in [Gir87]. An easier introduction is [Gir95]. The calculus of structures is due to Alessio Guglielmi [Gug07, GS01]. The system SKS has first been presented by Kai Brünnler and Alwen Tiu [BT01, Brü03].

2 Properties of deep inference

Some properties that immediately stand out when we naively look at deep inference proof systems are their *regularity*, *duality* and *atomicity*. In this section we will present these significant properties of systems that are only made possible because of the ability to apply rules deep inside of a formula.

To illustrate these properties, we will observe different deep inference systems that share the same characteristic features. The first of these is system SKS (Figure 11) for classical logic, introduced in the precious section. The second system we will consider is system SLLS (Figure 12): it is a sound and complete deep inference system for linear logic (see Section 1.3) which we will use as an example throughout these lecture notes. Last, we will see system BV (Figure 14), which features a self-dual non commutative connective $\lhd$. We feature this system since it is not possible to support these connectives in shallow systems such as the sequent calculus.

2.1 Locality and atomicity

A main feature of deep inference is that rules can be made *local*, in the sense that determining whether an application of the rule is correct we do not need to inspect arbitrarily big formulae. This is achieved by turning all structural rules such as contraction and cut into their atomic versions: as we can see, in systems SKS, SLLS and BV all the structural rules only concern atoms. Atomicity is only possible in systems where we can apply rules deep: restricting contractions to their atomic form in shallow systems like the sequent calculus is impossible.

Definition 2.1.1. A rule is *local* if it does not require the inspection of expressions of arbitrary size.

Consider the following rule instances

$$\mathsf{cont}\,\frac{\Gamma \vdash \Phi, A, A}{\Gamma \vdash \Phi, A} \qquad \text{vs.} \qquad \mathsf{ac}{\downarrow}\,\frac{S\{a \vee a\}}{\{a\}}$$

$$\mathsf{id}\,\frac{}{A \vdash A} \qquad \text{vs.} \qquad \mathsf{ai}{\downarrow}\,\frac{S\{\mathsf{t}\}}{S\{a \vee \bar{a}\}}$$

In the sequent rules, going from bottom to top in constructing a proof e.g., during proof search, through an instance of contraction a formula A of unbounded size is duplicated. Whatever mechanism performs this duplication, needs to inspect all of A, so it has to have a global view on A; whereas inspection in the atomic case is restricted to a single atom. Likewise, to inspect the correctness of the sequent identity rule, we need to verify that formulae on either side of the sequent are identical, which requires inspection of formulae of unbounded size. In the atomic case, we simply need to check that both atoms are the same.

Locality has a number of advantages in terms of proof normalisation (e.g. cut-elimination), proof system design, and generality. Having atomic rules allows us to follow atoms from their creation to their destruction. This ability is crucial for some graphical representations of proofs, as we will see in Section 5. Furthermore, locality allows for a normalisation procedure known as *decomposition*, which is not achievable in shallow proof systems. Decomposition procedures consist in restricting certain rules to specific parts of a derivation. For example, a

$$\mathsf{ai}{\downarrow}\,\frac{S\{1\}}{S\{a \parr \bar{a}\}} \qquad \mathsf{ai}{\uparrow}\,\frac{S\{a \otimes \bar{a}\}}{S\{\bot\}}$$

$$\mathsf{d}{\downarrow}\,\frac{S\{(A \parr B) \with (C \parr D)\}}{S\{(A \with C) \parr (B \oplus D)\}} \qquad \mathsf{d}{\uparrow}\,\frac{S\{(A \oplus B) \otimes (C \with D)\}}{S\{(A \otimes C) \oplus (B \otimes D)\}}$$

$$\mathsf{p}{\downarrow}\,\frac{S\{!(R \parr T)\}}{S\{!R \parr ?T\}} \qquad \mathsf{p}{\uparrow}\,\frac{S\{!R \otimes ?T\}}{S\{?(R \otimes T)\}}$$

$$\mathsf{aw}{\downarrow}\,\frac{S\{0\}}{S\{a\}} \qquad \mathsf{ac}{\downarrow}\,\frac{S\{a \oplus a\}}{S\{a\}} \qquad \mathsf{ac}{\uparrow}\,\frac{S\{a\}}{S\{a \with a\}} \qquad \mathsf{aw}{\uparrow}\,\frac{S\{a\}}{S\{\top\}}$$

$$\mathsf{nm}{\downarrow}\,\frac{S\{0\}}{S\{0 \with 0\}} \qquad \mathsf{s}\,\frac{S\{(A \parr B) \otimes C\}}{S\{(A \otimes C) \parr B\}} \qquad \mathsf{m}\,\frac{S\{(A \with B) \oplus (C \with D)\}}{S\{(A \oplus C) \with (B \oplus D)\}} \qquad \mathsf{nm}{\uparrow}\,\frac{S\{\top \oplus \top\}}{S\{\top\}}$$

$$\mathsf{nm}_1{\downarrow}\,\frac{S\{0\}}{S\{0 \parr 0\}} \qquad \mathsf{m}_1{\downarrow}\,\frac{S\{(A \parr B) \oplus (C \parr D)\}}{S\{(A \oplus C) \parr (B \oplus D)\}} \qquad \mathsf{m}_1{\uparrow}\,\frac{S\{(A \with B) \otimes (C \with D)\}}{S\{(A \otimes C) \with (B \otimes D)\}} \qquad \mathsf{nm}_1{\uparrow}\,\frac{S\{\top \otimes \top\}}{S\{\top\}}$$

$$\mathsf{nm}_2{\downarrow}\,\frac{S\{0\}}{S\{0 \otimes 0\}} \qquad \mathsf{m}_2{\downarrow}\,\frac{S\{(A \otimes B) \oplus (C \otimes D)\}}{S\{(A \oplus C) \otimes (B \oplus D)\}} \qquad \mathsf{m}_2{\uparrow}\,\frac{S\{(A \with B) \parr (C \with D)\}}{S\{(A \parr C) \with (B \parr D)\}} \qquad \mathsf{nm}_2{\uparrow}\,\frac{S\{\top \parr \top\}}{S\{\top\}}$$

$$\mathsf{nl}_1{\downarrow}\,\frac{S\{0\}}{S\{?0\}} \qquad \mathsf{l}_1{\downarrow}\,\frac{S\{?R \oplus ?T\}}{S\{?(R \oplus T)\}} \qquad \mathsf{l}_1{\uparrow}\,\frac{S\{!(R \with T)\}}{S\{!R \with !T\}} \qquad \mathsf{nl}_1{\uparrow}\,\frac{S\{!\top\}}{S\{\top\}}$$

$$\mathsf{nl}_2{\downarrow}\,\frac{S\{0\}}{S\{!0\}} \qquad \mathsf{l}_2{\downarrow}\,\frac{S\{!R \oplus !T\}}{S\{!(R \oplus T)\}} \qquad \mathsf{l}_2{\uparrow}\,\frac{S\{?(R \with T)\}}{S\{?R \with ?T\}} \qquad \mathsf{nl}_2{\uparrow}\,\frac{S\{?\top\}}{S\{\top\}}$$

$$\mathsf{nz}{\downarrow}\,\frac{S\{\bot\}}{S\{?0\}} \qquad \mathsf{z}{\downarrow}\,\frac{S\{?R \parr T\}}{S\{?(R \oplus T)\}} \qquad \mathsf{z}{\uparrow}\,\frac{S\{!(R \with T)\}}{S\{!R \otimes T\}} \qquad \mathsf{nz}{\uparrow}\,\frac{S\{!\top\}}{S\{1\}}$$

Figure 12: System SLLS

$$\begin{array}{ccc}
A \otimes B = B \otimes A & (A \otimes B) \otimes C = A \otimes (B \otimes C) & A \otimes 1 = A \\
A \with B = B \with A & (A \with B) \with C = A \otimes (B \with C) & A \with \top = A \\
A \oplus B = B \oplus A & (A \oplus B) \oplus C = A \oplus (B \oplus C) & A \oplus 0 = A \\
A \parr B = B \parr A & (A \parr B) \parr C = A \parr (B \parr C) & A \parr 1 = \bot \\
??R = ?R & !!R = !R & \\
\bot \oplus \bot = \bot = ?\bot & 1 \with 1 = 1 = !1 &
\end{array}$$

Figure 13: Equational theory of SLLS

$$\mathsf{ai}\!\downarrow \frac{S\{\circ\}}{S\{a \parr \bar{a}\}} \qquad \mathsf{ai}\!\uparrow \frac{S\{a \otimes \bar{a}\}}{S\{\circ\}}$$

$$\mathsf{s}\,\frac{S\{(A \parr B) \otimes C)\}}{S\{(A \otimes C) \parr B\}}$$

$$\mathsf{q}\!\downarrow \frac{S\{(A \parr B) \lhd (C \parr D)\}}{S\{(A \lhd C) \parr (B \lhd D)\}} \qquad \mathsf{q}\!\uparrow \frac{S\{(A \lhd B) \otimes (C \lhd D)\}}{S\{(A \otimes C) \lhd (B \otimes D)\}}$$

Figure 14: System SBV

$$\begin{array}{ccc} A \otimes B = B \otimes A & (A \otimes B) \otimes C = A \otimes (B \otimes C) & A \otimes \circ = A \\ A \parr B = B \parr A & (A \parr B) \parr C = A \parr (B \parr C) & A \parr \circ = A \\ & (A \lhd B) \lhd C = A \lhd (B \lhd C) & A \lhd \circ = A = \circ \lhd A \end{array}$$

Figure 15: Equational theory of SBV

useful decomposition procedure is one that allows us to constrain all instances of contraction to the bottom of a proof, and therefore to restrict all formula duplications to the bottom phase of the proof. Decomposition results provide new normal forms for proofs and an improved understanding of complexity creation during cut-elimination. We will expand on it in Sections 4.1 and 7.3. Last, atomicity introduces a strong regularity in the shape of inference rules, which is invaluable for generalising proof systems and obtaining general logic-agnostic properties, as we will see in Section 7.

Whereas reducing the identity and cut axioms to their atomic versions is possible in shallow systems such as the sequent calculus, the ability to apply rules deep inside of formulae is strictly necessary to make contraction atomic. The following counter-example is enough to show that.

Theorem 2.1.2. *The sequent*

$$\vdash a \wedge b, (\bar{a} \vee \bar{b}) \wedge (\bar{a} \vee \bar{b})$$

has no cut-free sequent proof in which all contractions are atomic.

Sketch of proof. We will show that no such proof exists in the one-sided sequent calculus of Figure 5 without the cut.

The idea is that we need to duplicate $a \wedge b$ to produce all the atoms necessary to close the proofs through identity axioms. However, as we are restricted to atomic contractions and we can only apply rules in a shallow way, we cannot apply contraction straight away since there are no single atoms.

We have to apply another rule, and it is easy to see that every rule we *can* apply separates the formulae into two independent branches in such a way that we get invalid premisses.

□

As we mentioned previously, a fundamental property for which we will exploit atomicity is *decomposition*, and in particular we are looking for the ability to restrict all contractions to the bottom of proofs. This decomposition is not possible without locality, which can be shown through a similar counter-example to the one above.

Theorem 2.1.3 (Decomposition)**.** *For every proof*

$$
\begin{array}{l}
\Vert\, \mathsf{SKS} \\
A
\end{array}
$$

in system SKS *there is a proof*

$$
\begin{array}{l}
\Vert\, \mathsf{SKS}\setminus\{\mathsf{ac}\!\downarrow\} \\
B \\
\Vert\, \{\mathsf{ac}\!\downarrow\} \\
A
\end{array}
$$

such that all instances of atomic contraction are restricted to the bottom of the proof. Furthermore, there is a cut-free proof

$$
\begin{array}{l}
\Vert\, \mathsf{SKS}\setminus\{\mathsf{ac}\!\downarrow,\mathsf{ai}\!\uparrow\} \\
C \\
\Vert\, \{\mathsf{ac}\!\downarrow\} \\
A
\end{array}
$$

such that all instances of atomic contraction are restricted to the bottom of the proof.

Example 2.1.4.

$$
\begin{array}{rc}
 & (\mathsf{t}\vee\mathsf{f})\wedge\mathsf{t} \\ \cline{2-2}
\mathsf{ai}\!\downarrow & (a\vee\bar{a}\vee\mathsf{f})\wedge\mathsf{t} \\ \cline{2-2}
\mathsf{aw}\!\downarrow & (a\vee a\vee\bar{a})\wedge\mathsf{t} \\ \cline{2-2}
\mathsf{ac}\!\downarrow & (a\vee\bar{a})\wedge\mathsf{t} \\ \cline{2-2}
\mathsf{ai}\!\downarrow & (a\vee\bar{a})\wedge(b\vee\bar{b}) \\ \cline{2-2}
\mathsf{s} & (a\wedge(b\vee\bar{b}))\vee\bar{a}
\end{array}
\quad\longrightarrow\quad
\begin{array}{rc}
 & (\mathsf{t}\vee\mathsf{f})\wedge\mathsf{t} \\ \cline{2-2}
\mathsf{ai}\!\downarrow & (a\vee\bar{a}\vee\mathsf{f})\wedge\mathsf{t} \\ \cline{2-2}
\mathsf{aw}\!\downarrow & (a\vee a\vee\bar{a})\wedge\mathsf{t} \\ \cline{2-2}
\mathsf{ai}\!\downarrow & (a\vee a\vee\bar{a})\wedge(b\vee\bar{b}) \\ \cline{2-2}
\mathsf{s} & ((a\vee a)\wedge(b\vee\bar{b}))\vee\bar{a} \\ \cline{2-2}
\mathsf{ac}\!\downarrow & (a\wedge(b\vee\bar{b}))\vee\bar{a}
\end{array}
$$

Note that this decomposition theorem holds specifically for *proofs* rather than for *derivations*. Recall that proofs are derivations with premiss t. Similar decomposition results exist for many systems. In this course we will show a few examples (Section 4.1) to illustrate the methodologies employed to prove such results, as well as introduce a generalised version of the above theorem (Section 7) to highlight just how general this phenomenon is in deep systems. Such decomposition is not achievable in shallow systems.

Theorem 2.1.5. *The sequent*

$$\vdash a\wedge a, \bar{a}\wedge\bar{a}$$

has no cut-free sequent proof in which all contractions are at the bottom.

Sketch of proof. We will show that no such proof exists in the one-sided sequent calculus of Figure 5 without the cut.

It suffices to show that, for any number of occurrences of the formulae $a\wedge a$ and $\bar{a}\wedge\bar{a}$ the sequent $\vdash a\wedge a,\dots,a\wedge a,\bar{a}\wedge\bar{a},\dots,\bar{a}\wedge\bar{a}$ is not provable without contraction.

We will show that every derivation without contraction with an endsequent of the form t $\vdash a \wedge a, \ldots, a \wedge a, \bar{a} \wedge \bar{a}, \ldots, \bar{a} \wedge \bar{a}$ has a leaf that contains at most a single atom. Since we need two single atoms to apply the identity axiom, such a derivation can never be a proof-. We will proceed by induction on the length of the derivation.

If the proof is of length 0, then it is obvious.

If the derivation Φ is of lenght n, let us choose a leaf l and remove its topmost rule ρ to obtain a derivation Φ'. By induction hypothesis, Φ' has a leaf that contains at most a single atom. If that leaf is not l, then Φ clearly also has a leaf that contains at most a single atom, and we are done. Otherwise, the conclusion of ρ contains at most a single atom. We have to show that in the premiss of ρ there is a sequent which contains at most a single atom. Since the connective $\vee$ does not occur in the endsequent, by the subformula property only the rules $\mathsf{id}, \mathsf{weak}, \mathsf{exch}$ and $\wedge$ can appera in the derivation.

- If ρ is an instance of $\mathsf{weak}\,\dfrac{\vdash \Gamma}{\vdash \Gamma, A}$, then $\vdash \Gamma$ contains at most a single atom-
- If ρ is an instance of $\mathsf{exch}\,\dfrac{\vdash \Delta, B, A, \Gamma}{\vdash \Delta, A, B, \Gamma}$, then $\vdash \Delta, A, B, \Gamma$ contains at most a single atom.
- If ρ is an instance of $\wedge\,\dfrac{\vdash \Gamma, A \qquad \vdash \Delta, B}{\vdash \Gamma, A \wedge B, \Delta}$, then Γ and Δ contain at most a single atom between the two of them, and thus one of them does not contain any single atoms. Without loss of generality, assume Δ does not contain any single atoms. Then, the sequent $\vdash \Delta, B$ contains at most a single atom.

□

The proofs of Theorems 2.1.2 and 2.1.5 only rely on the $\wedge$-introduction rule being *multiplicative*, i.e. of the form $\wedge\,\dfrac{\vdash \Gamma, A \qquad \vdash \Delta, B}{\vdash \Gamma, A \wedge B, \Delta}$ rather than $\wedge\,\dfrac{\vdash \Gamma, A \qquad \vdash \Gamma, B}{\vdash \Gamma, A \wedge B}$. For this reason, they can be replicated whether the shallow system is for propositional or for first-order predicate logic, whether it is two- or one-sided, whether or not rules for implication are in the system, whether weakening is explicit or buit into the axiom, whether a multiplicative or additive version of the $\vee$-introduction is used. Likewise, they can be adapted to non-classical logics with a multiplicative connective. In those shallow systems, contraction can thus neither be restricted to atoms nor to the bottom of a proof.

Exercise 2.1.6. Find SKS proofs of the sequents of Theorems 2.1.2 and 2.1.5 with all contractions restricted to the bottom.

2.2 Duality and Regularity

In deep inference systems every rule has its dual. We indicate the dual of a rule $\rho\!\downarrow$ by $\rho\!\uparrow$. The concept of dual rule very intuitively corresponds to the iadea of the *contrapositive* of an implication.

Since the premiss and conclusion of a deep inference rule are single formulas, dual rules can very easily be defined in through the negation of formulas. We can define the *negation* (or *dual*) of a formula inductively, using DeMorgan equivalences.

Example 2.2.1. In SKS, we inductively define the negation of a formula as:

$$\bar{\mathsf{t}} = \mathsf{f} \qquad \bar{\mathsf{f}} = \mathsf{t}$$
$$\overline{(A \vee B)} = \bar{A} \wedge \bar{B} \qquad \overline{(A \wedge B)} = \bar{A} \vee \bar{B}$$

Definition 2.2.2. The *dual of a rule* $\frac{S\{A\}}{S\{B\}}$ is the rule $\frac{S\{\bar{B}\}}{S\{\bar{A}\}}$.

Two rules are dual to each other if one can be obtained from the other by exchanging premise and conclusion and negating them. The duality between different rules is thereby exposed, like for example that of the identity axiom and the cut rule:

$$\mathsf{ai}\!\downarrow \frac{S\{\mathsf{t}\}}{S\{a \vee \bar{a}\}} \qquad \mathsf{ai}\!\uparrow \frac{S\{a \wedge \bar{a}\}}{S\{\mathsf{f}\}} \quad .$$

Some rules are said to be *self-dual*: the rule and its dual are identical. The medial and switch rules are self-dual:

$$\mathsf{m}\,\frac{S\{(A \wedge B) \vee (C \wedge D)\}}{S\{(A \vee C) \wedge (B \vee D)} \qquad \mathsf{s}\,\frac{S\{(A \vee B) \wedge C\}}{S\{(A \wedge C) \vee B\}}$$

We can extend the concept of dual rules to define dual derivations,as we will see in Section 3.

From naively observing the deep inference systems of these notes, we can quickly observe that deep inference rules present a striking regularity. Many of them are of the shape

$$\frac{S\{(A \,\alpha\, B) \,\beta\, (C \,\alpha'\, D)\}}{S\{(A \,\beta\, C) \,\alpha\, (B \,\beta'\, D)\}} \quad ,$$

where α, β, α' and β' are connectives. This shape is called in the *medial* shape.

In fact, we can make systems where *all* the rules have this shape, as we will see in Section 7. This regularity allows us to design systems in a systematic way, as well as to generalise proof-systems to reason about them independently of the logic.

2.3 Self-dual non commutative connectives

Consider multiplicative linear logic (see Section 1.3). If we enrich this system with a logical connective $\triangleleft$, which is self-dual ($\overline{A \triangleleft B} = \bar{A} \triangleleft \bar{B}$) and non-commutative ($A \triangleleft B \neq B \triangleleft A$), we obtain a logical system which to date is not known to be expressible in the sequent calculus.

In deep inference, this extension is expressible in the logical system BV [Gug07]. We know that, any restriction on the depth of the inference rules of the system would result in a strictly less expressive logical system[4]. We can therefore say that deep inference is a non-trivial extension to the traditional sequent calculus: it allows for a simple formulation of a logic which does not admit any straightforward formulation in sequent calculus without deep inference.

[4]This is proved by means of a counterexample in [Tiu06b]

2.4 Notes

System SLLS is due to Lutz Straßburger [Str03a]. System SBV is due to Alessio Guglielmi [Gug07]. The first definition of locality, as well as the counter-examples showing that locality and decomposition are impossible in the sequent calculus are by Kai Brünnler [Brü06].

$$\mathsf{ai}\!\downarrow \frac{S\{1\}}{S\{a \parr \bar{a}\}} \qquad \mathsf{s} \frac{S\{(A \parr B) \otimes C\}}{S\{(A \otimes C) \parr B\}} \qquad \mathsf{ai}\!\uparrow \frac{S\{a \otimes a\}}{S\{\bot\}}$$

Figure 16: System SMLS

$$\begin{array}{lll} A \otimes B = B \otimes A & (A \otimes B) \otimes C = A \otimes (B \otimes C) & A \otimes 1 = A \\ A \parr B = B \parr A & (A \parr B) \parr C = A \parr (B \parr C) & A \parr 1 = \bot \end{array}$$

Figure 17: Equational theory of SMLS

$$\mathsf{id} \frac{}{\vdash A, A^{\bot}} \qquad \mathsf{cut} \frac{\vdash A, \Phi \quad \vdash A^{\bot}, \Psi}{\vdash \Phi, \Psi}$$

$$\otimes \frac{\vdash A, \Phi \quad \vdash B, \Psi}{\vdash A \otimes B, \Phi, \Psi} \qquad \parr \frac{\vdash A, B, \Phi}{\vdash A \parr B, \Phi}$$

$$\bot \frac{\vdash \Phi}{\vdash \bot, \Phi} \qquad 1 \frac{}{\vdash 1}$$

Figure 18: Sequent calculus system MLL

3 Formalisms, Derivations and Proofs

In order to do proof theory, we need a concise definition of what constitutes a proof system, what is a derivation and what constitutes a proof. In deep inference we can apply rules at any depth inside the derivation, and therefore different formalisms are needed to capture this behaviour. In this section we will present two of these formalisms, and show how to obtain sequent proofs from deep inference proofs and viceversa, with the simple example of multiplicative linear logic.

Deep inference system SMLS for multiplicative linear logic is presented in Figure 16, and a sequent system is shown in Figure 18).

3.1 The Calculus of Structures

The calculus of structures is a deep inference formalism in which inference rules can be applied inside of a context $S\{\ \}$. Formulae are often considered modulo equality and rules are applied sequentially. Its motivations and some particularities have been informally introduced in Section 1.4, and in this section we will formally define *derivations* and *proofs*, in order to compare it to other deep inference formalisms.

Definition 3.1.1. An *inference rule* is a scheme of the form $\rho \frac{S\{R\}}{S\{T\}}$ where ρ is the *name* of the rule, R is its *premiss* and T its *conclusion*.

A *system* $\mathscr{S}$ is a set of inference rules.

In logical systems, we usually associate an *equational theory* to formulae. These may include associativity, commutativity and unit rules. Figures 10, 13 and 15 contain the equational theories for SKS, SLLS and BV respectively.

In the Calculus of Structures, we most frequently operate with formulas modulo equality: the equivalence class of equal formulas is called a *structure*. If for a specific purpose we do not operate modulo equality and instead incorporate equality rules directly as inference rules in the system (see Figure 11), we will state it explicitely. Note that unlike some sequent systems, most deep inference systems include units (like t and f) and we explicitly use the units in the introduction and cut rules.

Intuitively, given a set of rules, a straightforward way to define deep derivations is as a sequence of rules, each of which can be applied to any subformula of the conclusion of the previous rule.

Definition 3.1.2. A calculus of structures (CoS) derivation ϕ in $\mathscr{S}$ from *premiss* A to *conclusion* B denoted by $\begin{array}{c} A \\ \Phi \Vert \mathscr{S} \\ B \end{array}$ is defined to be one of the following:

- a formula $\Phi = A = B$;
- a *composition by inference*

$$\phi = \begin{array}{c} A \\ \Phi_1 \Vert \mathscr{S} \\ \rho\,\dfrac{S\{R\}}{S\{T\}} \\ \Phi_2 \Vert \mathscr{S} \\ B \end{array}$$

  where R and T are structures, $\rho\,\dfrac{R}{T}$ is an instance of an inference rule in $\mathscr{S}$ and Φ_1, Φ_2 are derivations in $\mathscr{S}$.

We denote by

$$\begin{array}{c} A \\ \Phi \Vert \{\rho_1,\ldots,\rho_n\} \\ B \end{array}$$

a derivation where only the rules $\rho_1, \ldots, \rho_n$ appear.

Sometimes we omit the name of a derivation or the name of the proof system if there is no ambiguity.

Definition 3.1.3. A *proof* is a derivation whose premiss is equal to a distinguished unit. We denote them by $\begin{array}{c} \Pi \Vert \mathscr{S} \\ A \end{array}$.

3.2 Open Deduction

As we saw in the previous section, a Calculus of Structures derivation is simply a sequence of rules. This allows us to order the rules, and to easily describe a rule as being *above* or *below* another rule. However, by applying rules sequentially, we *impose* this order, which can be superfluous in situations where the rule applications are completely independent from each other. Consider for example the following two derivations:

$$\cfrac{\cfrac{A \wedge B}{C \wedge B}}{C \wedge D} \qquad\qquad \cfrac{\cfrac{A \wedge B}{A \wedge D}}{C \wedge D} \quad . \tag{18}$$

The two inference rules applied are completely independent (each affecting only the red or blue part), and the order imposed on them is artificial: it is an example of what we call *bureaucracy*. When considering questions like the identity of proofs, we would like the two derivations above to be equivalent.

So far, we have seen throughout these notes that the motivation behind deep inference is the ability to apply rules at any depth. A straightforward consequence of that is in fact the ability to compose derivations vertically (through inference rules), but also horizontally through connectives:

$$\mathsf{s}\,\frac{(A \vee B) \wedge C}{(A \wedge C) \vee B} \wedge \mathsf{aw}{\downarrow}\,\frac{\mathsf{f}}{a} \quad .$$

Open Deduction derivations will be built as sequences of rules which can also be composed with the same connectives as we compose formulae. In this way, two Calculus of Structures derivations that are equivalent except for the order of application of two independent rules can be represented as the same Open Deduction derivation.

Definition 3.2.1. An open deduction (OD) derivation Φ in $\mathscr{S}$ from *premiss* A to *conclusion* B denoted by $\begin{array}{c} A \\ \phi\big\Vert\mathscr{S} \\ B \end{array}$ is defined to be one of the following:

- a formula $\Phi = A = B$;
- a *composition by inference*

$$\Phi = \rho\,\cfrac{\begin{array}{c} A \\ \Phi_1\big\Vert\mathscr{S} \\ A' \end{array}}{\begin{array}{c} B' \\ \Phi_2\big\Vert\mathscr{S} \\ B \end{array}}$$

where ρ is an instance of an inference rule in $\mathscr{S}$ and Φ_1 and Φ_2 are derivations in $\mathscr{S}$;

$$
\mathsf{ai}\!\downarrow \frac{\mathsf{t}}{a \vee \bar{a}} \qquad \mathsf{ai}\!\uparrow \frac{a \wedge \bar{a}}{\mathsf{f}}
$$

$$
\mathsf{s}\,\frac{(A \vee B) \wedge C}{(A \wedge C) \vee B} \qquad \mathsf{m}\,\frac{(A \wedge B) \vee (C \wedge D)}{(A \vee C) \wedge (B \vee D)}
$$

$$
\mathsf{ac}\!\downarrow \frac{a \vee a}{a} \qquad \mathsf{ac}\!\uparrow \frac{a}{a \wedge a}
$$

$$
\mathsf{aw}\!\downarrow \frac{\mathsf{f}}{a} \qquad \mathsf{aw}\!\uparrow \frac{a}{\mathsf{t}}
$$

Figure 19: Open Deduction presentation of SKS

- a *composition by connectives*

$$
\phi = \begin{array}{c} A_1 \\ \Phi_1 \Big\| \mathscr{S} \\ B_1 \end{array} \star \begin{array}{c} A_2 \\ \Phi_2 \Big\| \mathscr{S} \\ B_2 \end{array}
$$

  where $\star$ is a logical connective, $A \equiv A_1 \star A_2$, $B \equiv B_1 \star B_2$, and ϕ_1 and ϕ_2 are derivations in $\mathscr{S}$.

We can therefore represent the two derivations above (equation 18) as one single Open Deduction derivation:

$$
\dfrac{\dfrac{{\color{red}A} \wedge B}{{\color{red}C} \wedge {\color{blue}B}}}{C \wedge {\color{blue}D}} \quad \longrightarrow \quad \frac{A}{C} \wedge \frac{B}{D} \quad \longleftarrow \quad \dfrac{\dfrac{A \wedge {\color{blue}B}}{{\color{red}A} \wedge {\color{blue}D}}}{{\color{red}C} \wedge D}
$$

Since when we apply rules deep we are no longer applying them inside a context but rather performing a composition by connectives, we often describe open deduction systems without the context $S\{\ \}$ that is included in Calculus of Structures presentations, as in Figure 3.2. The same rules describe the same logics in both formalisms, but note that the equational theory is slightly different. The equations $\mathsf{f} \wedge \mathsf{f} = \mathsf{f}$ and $\mathsf{t} \vee \mathsf{t} = \mathsf{t}$ have been added as àrt of the equational theory (i.e. we wok modulo these equations as well), whereas in the CoS presentation they could be recovered through

$$
\mathsf{nm}\!\downarrow \frac{S\{\mathsf{f}\}}{S\{\mathsf{f} \wedge \mathsf{f}\}} \quad \text{and} \quad \dfrac{\mathsf{m}\,\dfrac{= \dfrac{S\{\mathsf{f} \wedge \mathsf{f}\}}{S\{\mathsf{f} \wedge ((\mathsf{f} \wedge \mathsf{t}) \vee (\mathsf{t} \wedge \mathsf{f}))\}}}{S\{\mathsf{f} \wedge ((\mathsf{f} \vee \mathsf{t}) \wedge (\mathsf{f} \vee \mathsf{t}))\}}}{S\{\mathsf{f}\}} \quad .
$$

Exercise 3.2.2. Show that for every open deduction derivation there is (at least) one calculus of structures derivation, and for every calculus of structures derivation, there is an open deduction derivation.

$$
\begin{array}{ccc}
A \wedge (B \wedge C) = (A \wedge B) \wedge C & A \wedge B = B \wedge A & A \wedge \mathsf{t} = A \\
A \vee (B \vee C) = (A \vee B) \vee C & A \vee B = B \vee A & A \vee \mathsf{f} = A \\
\mathsf{t} \vee \mathsf{t} = \mathsf{t} & \mathsf{f} \wedge \mathsf{f} = \mathsf{f} &
\end{array}
$$

Figure 20: Equational theory for the Open Deduction presentation of SKS

Having several formalisms allows us to choose the most convenient one depending on our goals. The sequential form of the Calculus of Structures makes it ideal for reasoning by induction, proofs where we need to look at the ordering of rules... On the other hand, the ability to apply rules in a synchronous way in Open Deduction makes it concise and ideal for reasoning about composition, about local rewritings, and in general about operations where proofs remain in the same *equivalence class* modulo superflous ordering of rules.

3.3 Operations with derivations

For both formalisms, we can define operations pn derivations, yielding new derivations. These operations will be used often when proving properties of systems.

Definition 3.3.1. Given a derivation $\begin{array}{c} A \\ \Phi \| \\ B \end{array}$, we define its dual $\begin{array}{c} \bar{B} \\ \bar{\Phi} \| \\ \bar{A} \end{array}$ as the derivation obtained by taking the dual of each inference rule in Φ.

Definition 3.3.2. Let $\begin{array}{c} A \\ \Phi \| \\ B \end{array}$ be a derivation, and $K\{\ \}$ a context. We define the derivation $K\{\Phi\}$ from $K\{A\}$ to $K\{B\}$ as the derivation obtained by inserting Φ in the place of the hole in $K\{\ \}$.

3.4 From deep inference to the sequent calculus and back

We will present the correspondence between sequent calculus proofs and deep inference proofs for the simple example of multiplicative linear logic (MLL). For this, we will consider the systems of Figures 16 and 18.

We want to construct a sequent proof for every CoS proof, and viceversa, with "the same" conclusion. To define such a concept, we introduce a translation from sequents to deep inference formulae, and conversely.

Definition 3.4.1. We consider the obvious translation between MLL formulae and SMLS formulae, in which propositional variables correspond to atoms, negated variables correspond to negated atoms, and compositions wih a connective correspond to the equivalent compositions via the same connective.

We extend this translation to sequents by setting that the empty sequent $\vdash$ corresponds to the formula $\perp$ and the sequent $\vdash A_1, \ldots, A_h$ corresponds to the formula $A_1 \parr \ldots \parr A_h$. We will abuse notation, and denote the formula obtained as a disjunction of all the formulae of a sequent Γ as Γ.

All that is needed to show the correspondence between proofs in the two formalisms are the following two lemmas.

Lemma 3.4.2. *For every rule* $\rho\,\dfrac{R}{T}$ *in* SMLS*, the sequent* $\vdash R^{\perp}, T$ *is provable in* MLL*.*

Proof. It is enough to show that it holds for the down-rules, since we can use the same corresponding MLL proofs for the dual rules.

For the rule $\mathsf{ai}{\downarrow}\,\dfrac{1}{a \parr \bar{a}}$ and its dual $\mathsf{ai}{\uparrow}\,\dfrac{a \parr \bar{a}}{\perp}$, we give the MLL proof

$$\perp\dfrac{\parr\dfrac{\mathsf{id}\dfrac{}{\vdash a, a^{\perp}}}{\vdash a \parr a^{\perp}}}{\vdash \perp, a \parr a^{\perp}} .$$

For the rule $\mathsf{s}\,\dfrac{(A \parr B) \otimes C}{(A \otimes C) \parr B}$, we give the MLL proof

$$\parr\dfrac{\parr\dfrac{\otimes\dfrac{\mathsf{id}\dfrac{}{\vdash C, C^{\perp}} \qquad \otimes\dfrac{\mathsf{id}\dfrac{}{\vdash A, A^{\perp}} \qquad \mathsf{id}\dfrac{}{\vdash B, B^{\perp}}}{\vdash A, (A^{\perp} \otimes B^{\perp}), B}}{\vdash (A^{\perp} \otimes B^{\perp}), C^{\perp}, (A \otimes C), B}}{\vdash (A^{\perp} \otimes B^{\perp}), C^{\perp}, (A \otimes C) \parr B}}{\vdash (A^{\perp} \otimes B^{\perp}) \parr C^{\perp}, (A \otimes C) \parr B} .$$

□

Lemma 3.4.3. *For any context* $S\{\ \}$*, if* $\vdash R^{\perp}, T$ *is provable in* MLL*, then* $\vdash S\{R\}^{\perp}, S\{T\}$ *is provable as well.*

Exercise 3.4.4. Prove the above Lemma.

Armed with these two lemmas, we proceed to show the translation from SMLS to MLL.

Theorem 3.4.5. *If there is a proof of* C *in* SMLS*, then* $\vdash C$ *is provable in* MLL*.*

Proof. We proceed by induction on the length of the proof Π of C.

In the base case, if $\Pi = 1$, then the corresponding MLL proof is $1\,\dfrac{}{\vdash 1}$.

Otherwise, $\Pi = \rho\,\dfrac{\overset{\|}{S\{R\}}}{S\{T\}}$. By Lemmas 3.4.2 and 3.4.3, the sequent $\vdash S\{R\}^{\perp}, S\{T\}$ is provable. By the induction hypothesis, the sequent $\vdash S\{R\}$ is provable.

We then get a proof of $\vdash S\{T\}$ through the cut rule:

$$\mathsf{cut}\,\dfrac{\vdash S\{R\} \qquad \vdash S\{R\}^{\perp}, S\{T\}}{\vdash S\{T\}} .$$

□

Conversely, we can translate from MLL to SMLS

Theorem 3.4.6. *If the sequent $\vdash \Gamma$ is provable in MLL, then Γ is provable in SMLS. Furthermore, if a sequent is cut-free provable in MLL, then its translation is cut-free provable in SMLS.*

Proof. We proceed by induction on the length of the proof Π of $\vdash \Gamma$.

If Π is $\mathsf{id}\,\dfrac{}{\vdash A, A^{\perp}}$, the corresponding proof $\begin{array}{c} 1 \\ \Big\Vert \{\mathsf{ai}\!\downarrow,\mathsf{s}\} \\ A \parr \bar{A} \end{array}$ is obtained as in Proposition 1.4.1.

If Π is $1\,\dfrac{}{\vdash 1}$, the corresponding proof is simply 1.

For the inductive step, we consider the last rule of Π.

- If the last rule is $\mathsf{cut}\,\dfrac{\vdash A, \Delta \quad \vdash A^{\perp}, \Theta}{\vdash \Delta, \Theta}$, then by induction hypothesis there are proofs $\begin{array}{c} 1 \\ \Phi_1 \Big\Vert \\ A \parr \Delta \end{array}$ and $\begin{array}{c} 1 \\ \Phi_2 \Big\Vert \\ \bar{A} \parr \Theta \end{array}$. We then construct the proof

$$\begin{array}{c} 1 \\ \Phi_1 \Big\Vert \\ A \parr \Delta \\ (A \parr \Delta) \otimes \Phi_2 \Big\Vert \\ \mathsf{2s}\,\dfrac{(A \parr \Delta) \otimes (\bar{A} \parr \Theta)}{\mathsf{ai}\!\uparrow \dfrac{(A \otimes \bar{A}) \parr \Delta \parr \Theta}{\Delta \parr \Theta}} \end{array} .$$

- If the last rule is $\otimes\,\dfrac{\vdash A, \Delta \quad \vdash B, \Theta}{\vdash A \otimes B, \Delta, \Theta}$, then by induction hypothesis there are proofs $\begin{array}{c} 1 \\ \Phi_1 \Big\Vert \\ A \parr \Delta \end{array}$ and $\begin{array}{c} 1 \\ \phi_2 \Big\Vert \\ B \parr \Theta \end{array}$. We then construct the proof

$$\begin{array}{c} 1 \\ \Phi_1 \Big\Vert \\ A \parr \Delta \\ (A \parr \Delta) \otimes \Phi_2 \Big\Vert \\ \mathsf{2s}\,\dfrac{(A \parr \Delta) \otimes (B \parr \Theta)}{(A \otimes B) \parr \Delta \parr \Theta} \end{array} .$$

- If the last rule is $\parr\,\dfrac{\vdash A, B, \Delta}{\vdash A \parr B, \Delta}$, then by induction hypothesis there is a proof $\begin{array}{c} 1 \\ \Phi \Big\Vert \\ A \parr B \parr \Delta \end{array}$.

- If the last rule is $\perp \frac{\vdash \Delta}{\vdash \perp, \Delta}$, then by induction hypothesis there is a proof $\begin{array}{c} 1 \\ \Phi \Big\| \\ \perp \parr \Delta \end{array}$.

Since cuts in the deep inference proofs are only introduced by the presence of cut in the sequent proofs, it is easy to see that from a cut-free sequent proof we would only obtain a cut-free deep inference proof. $\square$

These correspondence theorems, combined with cut-elimination for the sequent calculus (as in Theorem 1.3.7) immediately entail cut-elimination for the Calculus of Structures: given a CoS proof, we can translate it to a sequent proof, obtain a cut-free sequent proof of it, and translate it again into a cut-free CoS proof. However, to gain a full advantage of a formalism, it should have *internal* cut-elimination procedures: otherwise it cannot provide any advantage in terms of cut-elimination (for example in terms of the procedure, or of the complexity). In the next section, we will present such an internal cut-elimination procedure for deep inference systems.

3.5 Derivations of switch and medial

The two rules switch and medial of SKS

$$\mathsf{s}\frac{S\{A \wedge (B \vee C)\}}{S\{(A \wedge B) \vee C\}} \qquad \text{and} \qquad \mathsf{m}\frac{S\{(A \wedge B) \vee (C \wedge D)\}}{S\{(A \vee C) \wedge (B \vee D)\}} \tag{19}$$

have a particular shape; they do not introduce or delete subformulas: they only rearrange the structure of the formulas. In other words, they are *linear* rewriting rules. In this section, we study some properties of derivations that only use switch and medial.

First, we use the following convention for saving space: we use a new inference rule $=$, which stands for any derivation using only rules from the last two lines in Figure 11. More precisely,

$$= \frac{A}{B} \quad \text{abbreviates} \quad \begin{array}{c} A \\ \Big\| \{\alpha\downarrow,\alpha\uparrow,\sigma\downarrow,\sigma\uparrow,\mathsf{t}\downarrow,\mathsf{f}\uparrow,\mathsf{f}\downarrow,\mathsf{t}\uparrow\} \\ B \end{array} .$$

The following lemma states a very useful property of the switch. It allows to move an arbitrary formula into or out of an arbitrary context.

Lemma 3.5.1. *For any formulas R, T and context $S\{\ \}$, there exist derivations*

$$\begin{array}{c} R \wedge S\{T\} \\ \Big\| \{=,\mathsf{s}\} \\ S\{R \wedge T\} \end{array} \quad \textit{and} \quad \begin{array}{c} S\{R \vee T\} \\ \Big\| \{=,\mathsf{s}\} \\ S\{R\} \vee T \end{array} .$$

Proof. We proceed by structural induction on $S\{\ \}$.

If $S\{\ \} = \{\ \}$, it is obvious.

If $S\{T\} = S'\{T\} \vee Q$, by induction hypothesis there is a derivation Φ such that

$$\mathsf{s}\,\frac{R \wedge (S'\{T\} \vee Q)}{\begin{array}{c}(R \wedge S'\{T\}) \vee Q \\ \Phi \Big\| \{\mathsf{s},=\} \\ S'\{R \wedge T\} \vee Q\end{array}} \quad .$$

If $S\{T\} = S'\{T\} \wedge Q$, by induction hypothesis there is a derivation Φ such that

$$\begin{array}{c}R \wedge S'\{T\} \wedge Q \\ \Phi \Big\| \{\mathsf{s},=\} \\ S'\{R \wedge T\} \wedge Q\end{array} \quad .$$

□

The next Lemma follows immediately, by replacing some formulae with units.

Lemma 3.5.2. *Given a context $S\{\ \}$ and a formula A, there exist derivations*

$$\begin{array}{c}A \wedge S\{\mathsf{t}\} \\ \Big\| \{\mathsf{s},=\} \\ S\{A\}\end{array} \qquad \textit{and} \qquad \begin{array}{c}S\{A\} \\ \Big\| \{\mathsf{s},=\} \\ S\{\mathsf{f}\} \vee A\end{array} \quad .$$

An example of an application of this lemma is the following transformation of a derivation that takes a single instance of an $\mathsf{ai}{\downarrow}$ and puts it at the top of a derivation.

$$\begin{array}{c}A \\ \Phi' \Big\| \\ \mathsf{ai}{\downarrow}\,\dfrac{S\{\mathsf{t}\}}{S\{a \vee \bar{a}\}} \\ \Phi'' \Big\| \\ B\end{array} \quad \to \quad \begin{array}{c}\mathsf{t}{\downarrow}\,\dfrac{A}{\mathsf{t} \wedge A} \\ \mathsf{ai}{\downarrow}\,\overline{(a \vee \bar{a}) \wedge A} \\ (a\vee\bar{a})\wedge\Phi' \Big\| \\ (a \vee \bar{a}) \wedge S\{\mathsf{t}\} \\ \Big\| \{\mathsf{s},=\} \\ S\{a \vee \bar{a}\} \\ \Phi'' \Big\| \\ B\end{array} \quad , \tag{20}$$

We can perform the dual operation for $\mathsf{ai}{\uparrow}$.

Notice that the switch rule of linear logic, as seen in Figures 12 and 16 has the exact same behaviour. Identical Lemmas hold for SLLS and SMLS, since these properties are due to the shape of the rule, and not to the specific connectives.

Exercise 3.5.3. Show that in systems SLLS and SMLS, for any formulas R, T and context $S\{\ \}$, there exist derivations

$$\begin{array}{c}R \otimes S\{T\} \\ \Big\| \{=,\mathsf{s}\} \\ S\{R \otimes T\}\end{array} \qquad \text{and} \qquad \begin{array}{c}S\{R \parr T\} \\ \Big\| \{=,\mathsf{s}\} \\ S\{R\} \parr T\end{array} \quad .$$

Using Lemma 3.5.1 we can obtain the following derivation

$$\begin{array}{c} S\{\mathsf{t}\} \wedge T\{A\} \\ \Big\| \{\mathsf{s}\} \\ S\{A\} \vee T\{\mathsf{f}\} \end{array} \tag{21}$$

for any $S\{\ \}$, $T\{\ \}$ and A, by working inductively on the contexts $S\{\ \}$ and $T\{\ \}$. We can do this according to the following two schemes:

$$\begin{array}{c} S\{\mathsf{t}\} \wedge T\{A\} \\ \Big\| \{\mathsf{s}\} \\ T\{S\{\mathsf{t}\} \wedge A\} \\ \Big\| \{\mathsf{s}\} \\ T\{S\{A\}\} \\ \Big\| \{\mathsf{s}\} \\ S\{A\} \vee T\{\mathsf{f}\} \end{array} \qquad \text{and} \qquad \begin{array}{c} S\{\mathsf{t}\} \wedge T\{A\} \\ \Big\| \{\mathsf{s}\} \\ S\{T\{A\}\} \\ \Big\| \{\mathsf{s}\} \\ S\{A \vee T\{\mathsf{f}\}\} \\ \Big\| \{\mathsf{s}\} \\ S\{A\} \vee T\{\mathsf{f}\} \end{array} \quad .$$

This allows us to define the following 'macro' rule ss, called *super switch*, to be a shorthand for any derivation of the above form:

$$\mathsf{ss}\,\frac{S\{\mathsf{t}\} \wedge T\{A\}}{S\{A\} \vee T\{\mathsf{f}\}} \quad . \tag{22}$$

Example 3.5.4. For $S\{\ \} = (\{\ \} \vee b) \wedge c$ and $T\{\ \} = (d \wedge \{\ \}) \vee e$, we have

$$\begin{array}{l} \phantom{\mathsf{s}\ } ((\mathsf{t} \vee b) \wedge c) \wedge ((d \wedge A) \vee e) \\ = \overline{((\mathsf{t} \vee b) \wedge c) \wedge ((A \wedge d) \vee e)} \\ \mathsf{s}\ \overline{(((\mathsf{t} \vee b) \wedge c) \wedge (A \wedge d)) \vee e} \\ = \overline{(((A \wedge (\mathsf{t} \vee b)) \wedge c) \wedge d) \vee e} \\ \mathsf{s}\ \overline{((((A \wedge \mathsf{t}) \vee b) \wedge c) \wedge d) \vee e} \\ = \overline{(d \wedge (\mathsf{f} \vee ((A \vee b) \wedge c))) \vee e} \\ \mathsf{s}\ \overline{((d \wedge \mathsf{f}) \vee ((A \vee b) \wedge c)) \vee e} \\ = \overline{((A \vee b) \wedge c) \vee ((d \wedge \mathsf{f}) \vee e)} \end{array} \quad .$$

Another useful derivation is the following, which allows to go from a subformula $A \wedge B$ to $A \vee B$ without introducing or deleting any other atoms:

$$\Psi \quad = \quad \begin{array}{l} \phantom{\mathsf{m}\ } (A \vee B) \vee \mathsf{t} \\ = \overline{((A \wedge \mathsf{t}) \vee (\mathsf{t} \wedge B)) \vee \mathsf{t}} \\ \mathsf{m}\ \overline{((A \vee \mathsf{t}) \wedge (\mathsf{t} \vee B)) \vee \mathsf{t}} \\ = \overline{((A \vee \mathsf{t}) \wedge (B \vee \mathsf{t})) \vee \mathsf{t}} \\ \mathsf{s}\ \overline{(((A \vee \mathsf{t}) \wedge B) \vee \mathsf{t}) \vee \mathsf{t}} \\ = \overline{(B \wedge (A \vee \mathsf{t})) \vee \mathsf{t}} \\ \mathsf{s}\ \overline{((B \wedge A) \vee \mathsf{t}) \vee \mathsf{t}} \\ = \overline{(A \wedge B) \vee \mathsf{t}} \end{array} \tag{23}$$

With Ψ and ss together, we can 'move' an atom a from one context $S\{\ \}$ to another context $T\{\ \}$, again without without introducing or deleting any other atoms.[5]

$$\begin{array}{c} \mathsf{ss}\,\dfrac{(S\{\mathsf{t}\} \wedge T\{a\}) \vee \mathsf{t}}{(S\{a\} \vee T\{\mathsf{f}\}) \vee \mathsf{t}} \\ \Psi \Big\| \{\mathsf{s},\mathsf{m}\} \\ (S\{a\} \wedge T\{\mathsf{f}\}) \vee \mathsf{t} \end{array} .$$

This construction can be used zero or more times, for any $h \geq 0$:

$$\begin{array}{c} (S\{\mathsf{t}\} \cdots \{\mathsf{t}\} \wedge T\{a_1\} \cdots \{a_h\}) \vee \mathsf{t} \\ \Phi \Big\| \{\mathsf{s},\mathsf{m}\} \\ (S\{a_1\} \cdots \{a_h\} \wedge T\{\mathsf{f}\} \cdots \{\mathsf{f}\}) \vee \mathsf{t} \end{array} \tag{24}$$

3.6 Notes

The Calculus of Structures formalism is due Alessio Guglielmi, originally presented in [Gug07, GS01]. Open Deduction is due to Alessio Guglielmi, Tom Gundersen, and Michel Parigot, as introduced in [GGP10]. The proof of the correspondence between CoS and the sequent calculus is adapted from [GS01]. For classical logic, this has first been shown by Brünnler and Tiu in [BT01]. Similar proofs for more complex logics such as first-order classical logic and full linear logic can be found in [Brü06] and [Str02b] respectively.

[5] This will become relevant in Section 5.

4 Normalisation and cut elimination

4.1 Decomposition

In Section 2 you have seen that deep inference formalisms can have nice properties, like locality, atomicity, regularity, and duality. In this section, we will start to see what consequences these properties can have.

Let us begin with the simplest symmetric (self-dual) system in the calculus of structures, that we have seen: SMLS, shown in Figure 16. It contains only three inference rules, and every derivation in SMLS can be decomposed into three subderivations each using only one infernce rule. This is our first *decomposition theorem*:

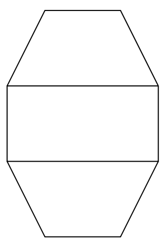
Theorem 4.1.1. *For every derivation* $\begin{array}{c} A \\ \Phi \Big\| \mathsf{SMLS} \\ B \end{array}$ *there is a derivation* $\begin{array}{c} A \\ \Big\| \{\mathsf{ai}\!\downarrow\} \\ A' \\ \Big\| \{\mathsf{s}\} \\ B' \\ \Big\| \{\mathsf{ai}\!\uparrow\} \\ B \end{array}$ *for some* A' *and* B'.

Proof. This can be proved by a simple permutation argument. We proceed by induction on the number of $\mathsf{ai}\!\downarrow$-instances, and consider the topmost one that is below a s- or $\mathsf{ai}\!\uparrow$-instance, and proceed by a second induction on the number of the s- and $\mathsf{ai}\!\uparrow$-instances that are above this $\mathsf{ai}\!\downarrow$. We permute this $\mathsf{ai}\!\downarrow$-instance now over the s or $\mathsf{ai}\!\uparrow$ immediately above is. There are only two types of cases to consider. Either we are in a situation as in (18), where permutation is trivial; or we have a case as shown on the left below

$$\mathsf{ai}\!\downarrow \dfrac{\mathsf{s}\,\dfrac{S\{(A\{1\} \parr C) \otimes B\}}{S\{(A\{1\} \otimes B) \parr C\}}}{S\{(A\{a \parr \bar{a}\} \otimes B) \parr C\}} \quad \leadsto \quad \mathsf{s}\,\dfrac{\mathsf{ai}\!\downarrow \dfrac{S\{(A\{1\} \parr C) \otimes B\}}{S\{(A\{a \parr \bar{a}\} \parr C) \otimes B\}}}{S\{(A\{a \parr \bar{a}\} \otimes B) \parr C\}}$$

which can be replaced by the derivation on the right above. The cases where the $\mathsf{ai}\!\downarrow$ acts inside the B or the C are similar. ☐

The decomposition that is obtained in Theorem 4.1.1 can be depicted as follows:



(25)

There is first a phase in which in every rule instance new material is created, then there is phase in which the size of the formula does not change, i.e., there is only rearrangement of material, and then there is a third phase in which every rule instance destroys material. This pattern reoccurs in many different decomposition theorems. We can state it immediately for System SKSg shown in Figure 9 (together with the equational theory in Figure 10):

Theorem 4.1.2. *For every derivation* $\Phi \left\| \begin{matrix} A \\ \mathsf{SKSg} \\ B \end{matrix} \right.$ *there is a derivation*

$$\begin{array}{c} A \\ \| \{\mathsf{ai}\!\downarrow,\mathsf{w}\!\downarrow,\mathsf{c}\!\uparrow\} \\ A' \\ \| \{\mathsf{s}\} \\ B' \\ \| \{\mathsf{ai}\!\uparrow,\mathsf{w}\!\uparrow,\mathsf{c}\!\downarrow\} \\ B \end{array}$$

for some A' *and* B'.

There is even a stronger form:

Theorem 4.1.3. *For every derivation* $\Phi \left\| \begin{matrix} A \\ \mathsf{SKSg} \\ B \end{matrix} \right.$ *there is a derivation*

$$\begin{array}{c} A \\ \| \{\mathsf{c}\!\uparrow\} \\ A' \\ \| \{\mathsf{w}\!\downarrow\} \\ A'' \\ \| \{\mathsf{ai}\!\downarrow\} \\ A''' \\ \| \{\mathsf{s}\} \\ B''' \\ \| \{\mathsf{ai}\!\uparrow\} \\ B'' \\ \| \{\mathsf{w}\!\uparrow\} \\ B' \\ \| \{\mathsf{c}\!\downarrow\} \\ B \end{array}$$

for some A', A'', A''', B''', B'', *and* B'.

Theorem 4.1.2 is an immediate corollary of Theorem 4.1.3. But the proof of Theorem 4.1.3 is much more complicated than the proof of Theorem 4.1.1. The reason is that the cocontraction rule

$$\mathsf{c}\!\uparrow \frac{S\{A\}}{S\{A \wedge A\}}$$

cannot as easily permuted up as the $\mathsf{ai}\!\downarrow$ in the proof of Theorem 4.1.1. To understand the problem, consider the simple situation on the left below, showing an instance of $\mathsf{c}\!\uparrow$ immediately below an instance of s:

$$\mathsf{c}\!\uparrow \dfrac{\mathsf{s}\,\dfrac{S\{((A \vee C) \wedge B) \vee D\}}{S\{(A \wedge B) \vee C \vee D\}}}{S\{(A \wedge B) \vee ((C \vee D) \wedge (C \vee D))\}} \quad \leadsto \quad \mathsf{c}\!\downarrow \dfrac{\mathsf{s}\,\dfrac{\mathsf{s}\,\dfrac{\mathsf{s}\,\dfrac{\mathsf{s}\,\dfrac{\mathsf{c}\!\uparrow \dfrac{S\{((A \vee C) \wedge B) \vee D\}}{S\{(((A \vee C) \wedge B) \vee D) \wedge (((A \vee C) \wedge B) \vee D)\}}}{S\{(((A \vee C) \wedge B) \vee D) \wedge ((A \wedge B) \vee C \vee D)\}}}{S\{((A \wedge B) \vee C \vee D) \wedge ((A \wedge B) \vee C \vee D)\}}}{S\{(A \wedge B) \vee (((A \wedge B) \vee C \vee D) \wedge (C \vee D))\}}}{S\{(A \wedge B) \vee (A \wedge B) \vee ((C \vee D) \wedge (C \vee D))\}}}{S\{(A \wedge B) \vee ((C \vee D) \wedge (C \vee D))\}} \tag{26}$$

The only way to permute this $\mathsf{c}\uparrow$ over the s is shown on the right above. The problem is not only that one switch is relaced by four instances of switch, but also that an new instance of $\mathsf{c}\downarrow$ is created that has to be permuted down in the next step, and which can in turn (by a dual argument) create new instances of $\mathsf{c}\uparrow$.

This is not simplified by reducing contraction to atomic form and removing the equational theory by making associativity and commutativity explicit, as done in the version of SKS (shown in Figure 11). Then the problematic case is when an $\mathsf{ac}\uparrow$ meets an $\mathsf{ai}\downarrow$:

$$\mathsf{ac}\uparrow\frac{\mathsf{ai}\downarrow\dfrac{S\{\mathsf{t}\}}{S\{a \vee \bar{a}\}}}{S\{(a \wedge a) \vee \bar{a}\}} \quad \rightsquigarrow \quad \mathsf{ac}\downarrow\frac{\alpha\downarrow,\sigma\downarrow\dfrac{\mathsf{s}\dfrac{\mathsf{ai}\downarrow\dfrac{\mathsf{t}\downarrow\dfrac{\mathsf{ai}\downarrow\dfrac{S\{\mathsf{t}\}}{S\{a \vee \bar{a}\}}}{S\{(a \wedge \mathsf{t}) \vee \bar{a}\}}}{S\{(a \wedge (a \vee \bar{a})) \vee \bar{a}\}}}{S\{((a \wedge a) \vee \bar{a}) \vee \bar{a}\}}}{S\{(a \wedge a) \vee (\bar{a} \vee \bar{a})\}}}{S\{(a \wedge a) \vee \bar{a}\}}$$

The instance of $\mathsf{ac}\uparrow$ disappears, but a new instance of $\mathsf{ac}\downarrow$ is created, which in turn can create a new instance of $\mathsf{ac}\uparrow$, and so on. In principle, we could be in a cyclic situation, contractions creating cocontractions endlessly, and vice versa. It is is not immediately obvious whether this process terminates or not.

Exercise 4.1.4. Take a guess. Does it terminate or not? Write your guess down. Later in the course we will give an answer.

Theorem 4.1.5. *For every derivation* $\begin{array}{c} A \\ \Phi\,\Big\| \mathsf{SKS} \\ B \end{array}$ *there is a derivation* $\begin{array}{c} A \\ \Big\| \{\mathsf{ac}\uparrow\} \\ A' \\ \Big\| \mathsf{SKS}\setminus\{\mathsf{ac}\downarrow,\mathsf{ac}\uparrow\} \\ B' \\ \Big\| \{\mathsf{ac}\downarrow\} \\ B \end{array}$ *for some* A' *and* B'.

Exercise 4.1.6. Prove Theorem 4.1.5. You can use Theorem 4.1.3 and what you have learned in Section 1.4.

Exercise 4.1.7. Use Theorem 4.1.3 to prove that for every derivation $\begin{array}{c} A \\ \Phi\,\Big\| \mathsf{SKSg} \\ B \end{array}$ there is a

$$\text{derivation}\quad \begin{array}{l} A \\ \Vert\,\{\mathsf{c}\!\uparrow\} \\ A' \\ \Vert\,\{\mathsf{ai}\!\downarrow\} \\ A'' \\ \Vert\,\{\mathsf{w}\!\downarrow\} \\ A''' \\ \Vert\,\{\mathsf{s}\} \\ B''' \\ \Vert\,\{\mathsf{w}\!\uparrow\} \\ B'' \\ \Vert\,\{\mathsf{ai}\!\uparrow\} \\ B' \\ \Vert\,\{\mathsf{c}\!\downarrow\} \\ B \end{array} \quad \text{for some } A',\ A'',\ A''',\ B''',\ B'',\ \text{and } B'.$$

All decompositions that we have discussed so far, followed the spirit of (25). This is also called the *first decomposition theorem*. This name suggests that there is a *second decomposition theorem*, which says that every derivation can be decomposed as follows:

$$\begin{array}{l} A \\ \Vert\,\text{nocore up} \\ A' \\ \Vert\,\{\mathsf{ai}\!\downarrow\} \\ A'' \\ \Vert\,\text{core (up and down)} \\ B'' \\ \Vert\,\{\mathsf{ai}\!\uparrow\} \\ B' \\ \Vert\,\text{noncore down} \\ B \end{array} \tag{27}$$

where the *core* is the part of the system that is needed to reduce the general interaction rules $\mathsf{i}\!\downarrow$ and $\mathsf{i}\!\uparrow$ to their atomic forms $\mathsf{ai}\!\downarrow$ and $\mathsf{ai}\!\uparrow$. For SKSg, shown in Figure 9, only the switch rule s is in the core. The second decomposition theorem for SKSg is therefore:

Theorem 4.1.8. *For every derivation* $\Phi \left\Vert{\scriptstyle \mathsf{SKSg}}\right. \begin{array}{c} A \\ \\ B \end{array}$ *there is a derivation*

$$
\begin{array}{c}
A \\
\Vert \{\mathsf{c}\uparrow,\mathsf{w}\uparrow\} \\
A' \\
\Vert \{\mathsf{ai}\downarrow\} \\
A'' \\
\Vert \{\mathsf{s}\} \\
B'' \\
\Vert \{\mathsf{ai}\uparrow\} \\
B' \\
\Vert \{\mathsf{c}\downarrow,\mathsf{w}\downarrow\} \\
B
\end{array}
$$

for some A', A'', B'', *and* B'.

There is, in fact, a stronger form:

Theorem 4.1.9. *For every derivation* $\Phi \left\Vert{\scriptstyle \mathsf{SKSg}}\right. \begin{array}{c} A \\ \\ B \end{array}$ *there is a derivation*

$$
\begin{array}{c}
A \\
\Vert \{\mathsf{c}\uparrow\} \\
A' \\
\Vert \{\mathsf{w}\uparrow\} \\
A'' \\
\Vert \{\mathsf{ai}\downarrow\} \\
A''' \\
\Vert \{\mathsf{s}\} \\
B''' \\
\Vert \{\mathsf{ai}\uparrow\} \\
B'' \\
\Vert \{\mathsf{w}\downarrow\} \\
B' \\
\Vert \{\mathsf{c}\downarrow\} \\
B
\end{array}
$$

for some A', A'', A''', B''', B'', *and* B'.

Proving the second decomposition via rule permutations is as problematic as for the first decomposition. Contraction poses the same problems as discussed in (26). Additionally, here weakening poses a similar problem, which is much less severe, as shown below:

$$
\mathsf{w}\uparrow \frac{\mathsf{s}\,\dfrac{S\{((A \vee C) \wedge B) \vee D\}}{S\{(A \wedge B) \vee C \vee D\}}}{S\{(A \wedge B) \vee \mathsf{t}\}}
\quad \rightsquigarrow \quad
\mathsf{w}\downarrow \frac{\mathsf{w}\uparrow\,\dfrac{S\{((A \vee C) \wedge B) \vee D\}}{S\{\mathsf{t}\}}}{S\{(A \wedge B) \vee \mathsf{t}\}}
\tag{28}
$$

In any case, we need different technology to prove these decomposition theorems, and we will come back to them later in the course.

Exercise 4.1.10. Derive Theorems 4.1.8 and 4.1.9 from each other.

The purpose of the second decomposition theorem is its usefuleness for proving cut elimination. Recall that in a self-dual deep inference system, the whole *up-fragment* corresponds to the *cut* and is admissible for *proofs*, i.e., derivations that start from t. If we have a proof

$$\begin{array}{c} \mathsf{t} \\ \Phi \Vert \mathsf{SKSg} \\ B \end{array}$$

then by Theorem 4.1.8 we get

$$\begin{array}{c} \mathsf{t} \\ \Vert \{\mathsf{ai}\downarrow\} \\ A'' \\ \Vert \{\mathsf{s}\} \\ B'' \\ \Vert \{\mathsf{ai}\uparrow\} \\ B' \\ \Vert \{\mathsf{c}\downarrow,\mathsf{w}\downarrow\} \\ B \end{array}$$

because neither $\mathsf{c}\downarrow$ nor $\mathsf{w}\downarrow$ can have premise t. And since $\mathsf{w}\downarrow$ and $\mathsf{c}\downarrow$ are already at the bottom for the derivation, the cut elimination now only has to be concerned with the rules $\mathsf{ai}\downarrow$, s, and $\mathsf{ai}\uparrow$, which is just multiplicative linear logic. Speaking more generally, if we have the second decomposition, cut elimination is only needed for the *core* of the system. This is exaclty what splitting is about and will be discussed in the next section.

4.2 Splitting

Splitting is based on a simple idea: to show that an atomic cut involving a and $\bar{a}$ is admissible, we follow a and $\bar{a}$ to the top of the derivation to find two independent subderivations, the premisses of which contain the dual of a and the dual of $\bar{a}$ respectively. In this way we obtain two proofs that don't interact above the cut, that we can rearrange to get a new cut-free proof.

$$\dfrac{\boxed{\begin{array}{c} H_a \otimes \dfrac{1}{\bar{a} \parr a} \\ \Vert \\ K_a \parr a \end{array}} \otimes \boxed{\begin{array}{c} \dfrac{1}{\bar{a} \parr a} \otimes H_{\bar{a}} \\ \Vert \\ \bar{a} \parr K_{\bar{a}} \end{array}}}{K_a \parr \dfrac{a \otimes \bar{a}}{\bot} \parr K_{\bar{a}}} \quad \xrightarrow{\text{splitting}} \quad \dfrac{H_a \otimes \dfrac{1}{\bar{a} \parr a} \otimes H_{\bar{a}}}{\boxed{\begin{array}{c} H_a \otimes \bar{a} \\ \Vert \\ K_a \end{array}} \parr \boxed{\begin{array}{c} a \otimes H_{\bar{a}} \\ \Vert \\ K_{\bar{a}} \end{array}}}$$

Proofs of cut-elimination by splitting therefore rely on two main properties of a proof system: the *dualities* present in it to ensure that each of the independent subproofs contains the dual of an atom involved in the cut, and the *shape* of the linear rules ensuring that the two proofs remain independent above the cut. In the CoS, the cut rule is usually divided into several rules. Often, only one of these rules is infinitary, the atomic cut, but all the rules involved in making the cut rule atomic can be shown admissible. The splitting method allows us to prove the admissibility of these cut rules in linear systems. Used in conjunction with decomposition, it allows us to prove cut-elimination in many systems, not necessarily linear.

The core propositional connectives of linear logic are divided into additive and multiplicative connectives, exemplifying perfectly the behaviours we are highlighting in this course in terms of how they split the context above them. The introduction rules for the additive conjunction $\&$ and the multiplicative conjunction $\otimes$ are given in their sequent calculus presentation as follows (see Section 1.3):

$$\frac{\vdash A, \Gamma \quad \vdash B, \Gamma}{\vdash A \& B, \Gamma} \quad , \qquad \frac{\vdash A, \Gamma \quad \vdash B, \Delta}{\vdash A \otimes B, \Gamma, \Delta} \quad .$$

Reading bottom up, we see that the additive conjunction $\wedge$ requires a duplication of the context whereas the multiplicative conjunction $\otimes$ requires that the context be divided between its hypotheses. There is no communication between Γ and Δ in the proof above the tensor rule where they are united.

$$\begin{array}{c} \otimes\dfrac{\begin{array}{c}\Pi_1\\ \vdash A, \Gamma\end{array} \quad \begin{array}{c}\Pi_2\\ \vdash B, \Delta\end{array}}{\vdash A \otimes B, \Gamma, \Delta} \\ \Delta \\ \vdash F\{A \otimes B\}, \Sigma \end{array}$$

It is precisely this multiplicative rule shape that splitting hinges on. In the sequent calculus, the presence of a main connective allows us to know exactly which rules can be applied above a cut. In deep inference, this is not possible since any rule can be applied at any depth, and we therefore focus on the behaviour of the context around a cut to tackle cut-elimination. This allows us to have a better understanding of how the cut-elimination procedure changes the proof globally. If all the connectives of a system require a splitting of the context like the multiplicative tensor does, then we can keep track of exactly how the context around a connective behaves. This allows us to split a proof into independent subproofs above every rule, just like in the example above the proof is divided into Π_1 and Π_2 above the $\otimes$ introduction rule. Cut-elimination is then only a matter of rearranging the independent subproofs into a cut-free proof.

Multiplicative linear logic (MLL) is the fragment of linear logic comprising only the multiplicative connectives and their units. It is a very simple system in which every connective requires such a splitting of the context, and therefore ideal to provide an example of a proof of cut-elimination via splitting. In what follows we will present a proof of cut-elimination via splitting for MLL. The same methodology can be used to prove splitting theorems for a wide variety of logics.

We will present this proof in CoS proof system for multiplicative linear logic SMLS (Figure 16). The form of the statement follows the standard scheme for splitting theorems: it is divided in two results for ease of reading, called *shallow splitting* and *context reduction*. We will work in a system for SMLS with explicit rules for the equational theory, as presented in 21. In this system we will also use an alternative version of switch, given by s'. Working modulo the equality rules, we can obtain s' from the usual presentation of switch s and viceversa as follows:

$$\mathsf{s}\,\dfrac{\mathsf{s}\,\dfrac{(A \parr B) \otimes (C \parr D)}{(A \otimes (C \parr D)) \parr B}}{(A \otimes C) \parr (B \parr D)} \qquad \text{and} \qquad =\dfrac{\mathsf{s}'\,\dfrac{=\dfrac{(A \parr B) \otimes C}{(A \parr B) \otimes (C \parr \bot)}}{(A \otimes C) \parr (B \parr \bot)}}{(A \otimes C) \parr B} \quad .$$

This updated presentation of switch will allow us to have a simple induction measure to prove the splitting result.

By simple observation, we can notice that in $\mathsf{SMLS}^{\downarrow}$ the scope of the connective $\otimes$ only decreases when reading top to bottom. The widening scope of relations from bottom to top is the main property used to prove splitting. If we follow a particular instance of the tensor $\otimes$ through a proof, its scope will be at its widest in the premiss. Therefore, if we have a proof of $F\{A \otimes B\}$, we can follow $\otimes$ up in the proof to obtain two independent proofs $\begin{array}{c}\Pi_1 \Vert \\ Q_A\{A\}\end{array}$ and $\begin{array}{c}\Pi_2 \Vert \\ Q_B\{B\}\end{array}$.

$$\dfrac{\boxed{\begin{array}{c}\Pi_1 \Vert \\ A \parr K_1 \parr Q_1\end{array}} \otimes \boxed{\begin{array}{c}\Pi_2 \Vert \\ B \parr K_2 \parr Q_2\end{array}}}{\boxed{\begin{array}{c}(A \parr K_1) \otimes (B \parr K_2) \\ \Vert \\ (A \otimes B) \parr K_1 \parr K_2\end{array}} \parr Q_1 \parr Q_2}$$

If we do this for every occurrence of $\otimes$ in the conclusion of a proof, starting from the outermost, we obtain a series of subproofs independent from each other. This is the gist of the splitting theorem, and cut-elimination comes as a corollary, by showing that we are free to rearrange these independent subproofs in such a way that the cut is no longer necessary.

In what follows we will present the splitting theorem for $\mathsf{SMLS}^{\downarrow}$. Our induction measure will be the length of Π, but without counting the aoccurrences of equality rules for $\parr$.

Definition 4.2.1. Given a proof Φ in $\mathsf{SMLS}^{\downarrow}$, we define the $\parr$-*length* of Φ as the number of inference rules in Φ different from the equality rules for the associativity, commutativity and unit of $\parr$. We denote it by $|\Phi|$.

Theorem 4.2.2 (Shallow splitting)**.** *For all formulae A, B, C, if there is a proof Π of $(A \otimes B) \parr C$ in $\mathsf{SMLS}^{\downarrow}$, there exist Q_1, Q_2 and*

$$\begin{array}{c} Q_1 \parr Q_2 \\ \Phi \Vert \\ C \end{array} \quad , \quad \begin{array}{c}\Pi_1 \Vert \\ A \parr Q_1\end{array} \quad , \quad \begin{array}{c}\Pi_2 \Vert \\ B \parr Q_2\end{array}$$

such that $|\Pi_1| + |\Pi_2| \leq |\Pi|$.

$$\mathsf{ai}\!\downarrow\frac{S\{1\}}{S\{a \parr \bar{a}\}} \qquad \mathsf{s}'\frac{S\{(A \parr B) \otimes (C \parr D)\}}{S\{(A \otimes C) \parr (B \parr D)\}} \qquad \mathsf{ai}\!\uparrow\frac{S\{a \otimes a\}}{S\{\bot\}}$$

$$\sigma\!\uparrow\frac{S\{A \otimes B\}}{S\{B \otimes A\}} \qquad \alpha\!\uparrow\frac{S\{(A \otimes B) \otimes C\}}{S\{A \otimes (B \otimes C)\}} \qquad \bot\!\uparrow\frac{S\{A\}}{S\{A \otimes 1\}} \qquad 1\!\uparrow\frac{S\{A \otimes 1\}}{S\{A\}}$$

$$\sigma\!\downarrow\frac{S\{A \parr B\}}{S\{B \parr A\}} \qquad \alpha\!\downarrow\frac{S\{(A \parr B) \parr C\}}{S\{A \parr (B \parr C)\}} \qquad \bot\!\downarrow\frac{S\{A \parr \bot\}}{S\{A\}} \qquad 1\!\downarrow\frac{S\{A\}}{S\{A \parr \bot\}}$$

Figure 21: System SMLS with explicit equality rules.

Proof. Since we do not count the equality rules for $\parr$, we can easily combine them together in a meta-rule. We define $=_{\parr}$ as the equivalence relation defined by the axioms for the associativity, commutativity and unit of $\parr$. We will work modulo this equivalence relation.

Given a proof Π of $(A \otimes B) \parr C$ in $\mathsf{SMLS}^{\downarrow}$, we proceed by induction on $|\Pi|$.

If $|\phi| = 0$, then $(A \otimes B) \parr C =_{\parr} 1$. Then, either:

- $A = B = 1, C = \bot$ and we take

$$\Phi = \begin{array}{c}\| \\ \bot \parr \bot\end{array} \quad , \qquad \Pi_1 = 1 \parr \bot \quad , \quad \Pi_2 = 1 \parr \bot \quad ; \text{or}$$

- $A = \bot, B = C = 1$ and we take

$$\Phi = 1 \parr \bot \quad , \qquad \Pi_1 = \bot \parr 1 \quad , \quad \Pi_2 = 1 \parr \bot \quad ; \text{or}$$

- $B = \bot, A = C = 1$ and we proceed symmetrically.

If $|\Pi| = n > 0$, inspection of the rules provides us the following possible cases:

$$(1)\ \Pi =_{\parr} \boxed{\begin{array}{c}\Pi' \| \\ \rho\dfrac{(A' \otimes B) \parr C}{(A \otimes B) \parr C}\end{array}} \quad ;$$

$$(2)\ \Pi =_{\parr} \boxed{\begin{array}{c}\Pi' \| \\ \rho\dfrac{(A \otimes B') \parr C}{(A \otimes B) \parr C}\end{array}} \quad ;$$

$$(3)\ \Pi =_{\parr} \boxed{\begin{array}{c}\Pi' \| \\ \rho\dfrac{(A \otimes B) \parr C'}{(A \otimes B) \parr C}\end{array}} \quad ;$$

(4) $$\Pi =_{\parr} \begin{array}{c} \Pi' \,\Big\Vert \\ \boxed{\mathsf{s}'\,\dfrac{((A \parr C_1) \otimes (B \parr C_2)) \parr C_3}{(A \otimes B) \parr C_1 \parr C_2 \parr C_3}} \end{array} \quad \text{with } C =_{\parr} C_1 \parr C_2 \parr C_3 \;;$$

(5) $$\Pi =_{\parr} \begin{array}{c} \Pi' \,\Big\Vert \\ \boxed{\mathsf{s}'\,\dfrac{(((A \otimes B) \parr C_1) \otimes (C_2 \parr C_3)) \parr C_4}{(A \otimes B) \parr (C_1 \otimes C_2) \parr C_3 \parr C_4}} \end{array} \quad \text{with } C =_{\parr} (C_1 \otimes C_2) \parr C_3 \parr C_4 \;;$$

(6) $$\Pi =_{\parr} \begin{array}{c} \Pi' \,\Big\Vert \\ \boxed{\otimes_a\,\dfrac{(A_1 \otimes (A_2 \otimes B)) \parr C}{((A_1 \otimes A_2) \otimes B) \parr C}} \end{array} \quad \text{with } A =_{\parr} A_1 \otimes A_2;$$

(7) $$\Pi =_{\parr} \begin{array}{c} \Pi' \,\Big\Vert \\ \boxed{\otimes_c\,\dfrac{(A \otimes B) \parr C}{(B \otimes A) \parr C}} \end{array} \quad ;$$

(8) $$\Pi =_{\parr} \begin{array}{c} \Pi' \,\Big\Vert \\ \boxed{\dfrac{(((A \otimes B) \parr C_1) \otimes 1) \parr C_2}{(A \otimes B) \parr C_1 \parr C_2}} \end{array} \quad \text{with } C =_{\parr} C_1 \parr C_2 \;;$$

(9) $$\Pi =_{\parr} \begin{array}{c} \Pi' \,\Big\Vert \\ \boxed{\dfrac{A \parr C}{(A \otimes 1) \parr C}} \end{array} \quad \text{with } B =_{\parr} 1 \;;$$

(1) Since $|\Pi'| = n - 1$, we apply the induction hypothesis to Π'. There exist Q_1, Q_2 and

$$\begin{array}{c} Q_1 \parr Q_2 \\ \Phi \,\Big\Vert \\ C \end{array} \quad , \qquad \Pi_1 =_{\parr} \begin{array}{c} \Pi'_1 \,\Big\Vert \\ \boxed{\rho\,\dfrac{A' \parr Q_1}{A \parr Q_1}} \end{array} \quad , \quad \begin{array}{c} \Pi_2 \,\Big\Vert \\ B \parr Q_2 \end{array}$$

such that $|\Pi_1| + |\Pi_2| = |\Pi'_1| + |\Pi_2| + 1 \leq |\Pi'| + 1 = |\Pi|$.

(2) This case is analogous to (1).

(3) We apply the induction hypothesis to Π'. There exist Q_1, Q_2 and

$$\begin{array}{c} Q_1 \parr Q_2 \\ \psi' \,\Big\Vert \\ \boxed{\rho\,\dfrac{C'}{C}} \end{array} \quad , \quad \begin{array}{c} \Pi_1 \,\Big\Vert \\ A \parr Q_1 \end{array} \quad , \quad \begin{array}{c} \Pi_2 \,\Big\Vert \\ B \parr Q_2 \end{array}$$

such that $|\Pi_1| + |\Pi_2| \leq |\Pi'| \leq |\Pi|$.

(4) We apply the induction hypothesis to Π'. There exist Q'_1, Q'_2 and

$$\begin{array}{c} Q'_1 \parr Q_2 \\ {\scriptstyle \Phi'}\Big\Vert \\ C_3 \end{array} \quad , \quad \begin{array}{c} {\scriptstyle \Pi_1}\Big\Vert \\ A \parr C_1 \parr Q'_1 \end{array} \quad , \quad \begin{array}{c} {\scriptstyle \Pi_2}\Big\Vert \\ B \parr C_2 \parr Q'_2 \end{array}$$

such that $|\Pi_1| + |\Pi_2| \leq |\Pi'| \leq |\Pi|$.

We take $Q_1 = C_1 \parr Q'_1$ and $Q_2 = C_1 \parr Q'_2$, and we have

$$\Phi = \begin{array}{c} C_1 \parr C_2 \parr Q'_1 \parr Q'_2 \\ {\scriptstyle \Phi'}\Big\Vert \\ C_1 \parr C_2 \parr C_3 \end{array} .$$

(5) We apply the induction hypothesis to Π'. There exist Q'_1, Q'_2 and

$$\begin{array}{c} Q'_1 \parr Q'_2 \\ {\scriptstyle \Phi_1}\Big\Vert \\ C_4 \end{array} \quad , \quad \begin{array}{c} {\scriptstyle \Pi'_1}\Big\Vert \\ (A \otimes B) \parr C_1 \parr Q'_1 \end{array} \quad , \quad \begin{array}{c} {\scriptstyle \Pi'_2}\Big\Vert \\ C_2 \parr C_3 \parr Q'_2 \end{array}$$

such that $|\Pi'_1| + |\Pi'_2| \leq |\Pi'|$.

We apply the induction hypothesis to Π'_1. There exist Q_1, Q_2 and

$$\Phi =_{\parr} \begin{array}{c} {\scriptstyle \otimes_u}\, \dfrac{Q_1 \parr Q_2}{(Q_1 \parr Q_2) \otimes 1} \\ {\scriptstyle \Phi_2}\Big\Vert \\ (C_1 \parr Q'_2) \otimes 1 \\ {\scriptstyle \Pi'_2}\Big\Vert \\ {\scriptstyle \mathsf{s}'}\, \dfrac{(C_1 \parr Q'_2) \otimes (C_2 \parr C_3 \parr Q'_2)}{(C_1 \otimes C_2) \parr C_3 \parr Q'_1 \parr Q'_2} \\ {\scriptstyle \Phi_1}\Big\Vert \\ (C_1 \otimes C_2) \parr C_3 \parr C_4 \end{array} \quad , \quad \begin{array}{c} {\scriptstyle \Pi_1}\Big\Vert \\ A \parr Q_1 \end{array} \quad , \quad \begin{array}{c} {\scriptstyle \Pi_2}\Big\Vert \\ B \parr Q_2 \end{array}$$

such that $|\Pi_1| + |\Pi_2| \leq |\Pi'_1| \leq |\Pi|$.

(6) We apply the induction hypothesis to Π'. There exist Q'_1, Q'_2 and

$$\begin{array}{c} Q'_1 \parr Q'_2 \\ {\scriptstyle \Phi_1}\Big\Vert \\ C \end{array} \quad , \quad \begin{array}{c} {\scriptstyle \Pi'_1}\Big\Vert \\ A_1 \parr Q'_1 \end{array} \quad , \quad \begin{array}{c} {\scriptstyle \Pi'_2}\Big\Vert \\ (A_2 \otimes B) \parr Q'_2 \end{array}$$

such that $|\Pi'_1| + |\Pi'_2| \leq |\Pi'|$.

We apply the induction hypothesis to Π'_2. There exist M, Q_2 and

$$\begin{array}{c} M \parr Q_2 \\ {\scriptstyle \Phi_2}\Big\| \\ Q'_2 \end{array} \quad , \quad \begin{array}{c} {\scriptstyle \Theta_1}\Big\| \\ A_2 \parr M \end{array} \quad , \quad \begin{array}{c} {\scriptstyle \Theta_2}\Big\| \\ B \parr Q_2 \end{array}$$

such that $|\Theta_1| + |\Theta_2| \leq |\Pi'_2|$.

We take $Q_1 \equiv Q'_1 \parr M$ and $\Pi_2 = \Theta_2$ and

$$\psi =_{\parr} \begin{array}{c} Q'_1 \parr M \parr Q_2 \\ {\scriptstyle \Phi_2}\Big\| \\ Q'_1 \parr Q'_2 \\ {\scriptstyle \Phi_1}\Big\| \\ C \end{array} \quad , \quad \Pi_1 =_{\parr} \begin{array}{c} 1 \otimes 1 \\ {\scriptstyle \Pi'_1}\Big\| \\ (A_1 \parr Q'_1) \otimes 1 \\ {\scriptstyle \Theta_1}\Big\| \\ \mathsf{s}'\,\dfrac{(A_1 \parr Q'_1) \otimes (A_2 \parr M)}{(A_1 \otimes A_2) \parr Q'_1 \parr M} \end{array} .$$

We have:
$|\Pi_1| + |\Pi_2| = |\Pi'_1| + |\Theta_1| + 1 + |\Theta_2| \leq |\Pi'_1| + |\Pi'_2| + 1 \leq |\Pi'| + 1 = |\Pi|$.

(7) This case is clear.

(8) We apply the induction hypothesis to Π'. There exist Q'_1, Q'_2 and

$$\begin{array}{c} Q'_1 \parr Q'_2 \\ {\scriptstyle \Phi_1}\Big\| \\ C_2 \end{array} \quad , \quad \begin{array}{c} {\scriptstyle \Pi'_1}\Big\| \\ (A \otimes B) \parr C_1 \parr Q'_1 \end{array} \quad , \quad \begin{array}{c} {\scriptstyle \Pi'_2}\Big\| \\ 1 \parr Q'_2 \end{array}$$

such that $|\Pi'_1| + |\Pi'_2| \leq |\Pi'|$.

We apply the induction hypothesis to Π'_1. There exist Q_1, Q_2 and

$$\Phi =_{\parr} \begin{array}{c} {\otimes_u}{\downarrow}\,\dfrac{Q_1 \parr Q_2}{(Q_1 \parr Q_2) \otimes 1} \\ {\scriptstyle \Phi_2}\Big\| \\ (C_1 \parr Q'_1) \otimes 1 \\ {\scriptstyle \Pi'_2}\Big\| \\ \mathsf{s}'\,\dfrac{(C_1 \parr Q'_1) \otimes (1 \parr Q'_2)}{(C_1 \otimes 1) \parr Q'_1 \parr Q'_2} \\ {\otimes_u}{\uparrow}\,\dfrac{}{C_1 \parr Q'_1 \parr Q'_2} \\ {\scriptstyle \Phi_1}\Big\| \\ C_1 \parr C_2 \end{array} \quad , \quad \begin{array}{c} {\scriptstyle \Pi_1}\Big\| \\ A \parr Q_1 \end{array} \quad , \quad \begin{array}{c} {\scriptstyle \Pi_2}\Big\| \\ B \parr Q_2 \end{array}$$

such that $|\Pi_1| + |\Pi_2| \leq |\Pi'_1| \leq |\Pi'| \leq |\Pi|$.

(9) We take

$$\Phi =_{\mathbin{\rotatebox[origin=c]{180}{\&}}} =_{\mathbin{\rotatebox[origin=c]{180}{\&}}} \frac{C \parr \bot}{C} \quad , \qquad \Pi_1 \equiv \begin{array}{c} \Pi' \Big\Vert \\ A \parr C \end{array} \quad , \qquad \Pi_2 \equiv =_{\parr} \frac{1}{1 \parr \bot} \quad .$$

We have $|\Pi_1| + |\Pi_2| = |\Pi'| \leq |\Pi|$.

$\square$

Exercise 4.2.3. Think about what would happen in the proof above if we had a medial rule like in classical logic. What would happen in its induction case?

Shallow splitting tells us that from 'shallow' contexts where the main connective is $\parr$ we can follow occurrences of $\otimes$ and of the atoms up in the proof and obtain independent subproofs. We can now apply shallow splitting starting from the outermost occurrences of $\otimes$ or the atoms, and apply this process recursively on every subproof to obtain a series of nested subproofs that in a way make-up the original proof. We formalise this recursive process in the following theorem.

Definition 4.2.4. We say that a context $H\{\ \}$ is *provable* if $H\{1\} = 1$.

Theorem 4.2.5 (Context Reduction)**.** *For any formula A and any context S, given a proof* $\begin{array}{c} \Pi \Big\Vert \mathsf{SMLS}^{\downarrow} \\ S\{A\} \end{array}$ *there exist a provable context $H\{\ \}$, a formula K and derivations*

$$\begin{array}{c} \\ \Theta \Big\Vert \\ K \parr A \end{array} \quad , \qquad \begin{array}{c} H\{K \parr \{\ \}\} \\ \chi \Big\Vert \\ S\{\ \} \end{array} \quad .$$

Proof. Given a context $S\{\ \}$ we define its *height* as the number of instances of $\otimes$ that $\{\ \}$ is in the scope of. We denote it by $|S|$. We proceed by induction on $|S|$.

- If $S\{A\} =_{\parr} A \parr K$, it is clear.

- If $S\{A\} =_{\parr} (S'\{A\} \otimes L) \parr M$ (L and M can be units), we apply Theorem 4.2.2. There exist Q_1, Q_2 and

$$\begin{array}{c} Q_1 \parr Q_2 \\ \Phi \Big\Vert \\ M \end{array} \quad , \qquad \begin{array}{c} \Pi_1 \Big\Vert \\ S'\{A\} \parr Q_1 \end{array} \quad , \qquad \begin{array}{c} \Pi_2 \Big\Vert \\ L \parr Q_2 \end{array} \quad .$$

We apply the induction hypothesis to $S'\{A\} \parr Q_1$. There exist a provable context $H\{\ \}$, a formula K and derivations

$$\begin{array}{c} \\ \Theta \Big\| \\ K \parr A \end{array} \quad , \qquad \chi = \begin{array}{c} H\{K \parr \{\ \}\} \otimes 1 \\ \chi' \Big\| \\ (S'\{\ \} \parr Q_1) \otimes 1 \\ \Pi_2 \Big\| \\ \mathsf{s}' \dfrac{(S'\{\ \} \parr Q_1) \otimes (L \parr Q_2)}{(S'\{\ \} \otimes L) \otimes (Q_1 \parr Q_2)} \\ \Phi \Big\| \\ (S'\{\ \} \otimes L) \otimes M \end{array} \quad .$$

We take $H\{\ \} \equiv H'\{\ \} \otimes 1$.

□

The splitting results are stronger than cut-elimination: they give us information about the structure of a proof and the 'pieces' from which it's built. Cut-elimination is a corollary of these results, stemming from our ability to rearrange these pieces in a way that suits us and still obtain a proof.

Finally, we have all the pieces to show that we can eliminate cuts.

Corollary 4.2.6 (Cut Elimination)**.** *For any context S and any proof*

$$\Pi \equiv \begin{array}{c} \Big\| \mathsf{SMLS}{\downarrow} \\ S\left\{ \mathsf{ai}{\uparrow} \dfrac{a \otimes \bar{a}}{\bot} \right\} \end{array} \quad ,$$

there is a proof

$$\begin{array}{c} \Pi' \Big\| \mathsf{SMLS}{\downarrow} \\ S\{\bot\} \end{array} \quad .$$

Proof. Given a proof $\begin{array}{c} \Big\| \mathsf{SMLS}{\downarrow} \\ S\{a \otimes \bar{a}\} \end{array}$, we apply Theorem 4.2.5.

There exist a provable context H, a formula K and derivations

$$\begin{array}{c} \Theta \Big\| \mathsf{SMLS}{\downarrow} \\ K \parr (a \otimes \bar{a}) \end{array} \quad , \qquad \begin{array}{c} H\{K \parr \{\ \}\} \\ \chi \Big\| \\ S\{\ \} \end{array} \quad .$$

We apply Theorem 4.2.2 to ζ. There are formulae Q_1, Q_2 and derivations

$$\begin{array}{c} Q_1 \parr Q_2 \\ \Phi \Big\| \\ K \end{array} \quad , \qquad \begin{array}{c} \Pi_1 \Big\| \mathsf{SMLS}{\downarrow} \\ a \parr Q_1 \end{array} \quad , \qquad \begin{array}{c} \Pi_2 \Big\| \mathsf{SMLS}{\downarrow} \\ \bar{a} \parr Q_2 \end{array} \quad .$$

If we isolate the first occurrence of the atom appearing in their conclusion, Π_1 and Π_2 have to be of the form

$$
\Pi_1 = \begin{array}{c} \Pi'_1 \Big\| \\ \mathsf{ai}{\downarrow}\dfrac{J_1\{1\}}{J_1\{a \parr \bar{a}\}} \\ \Psi_1 \Big\| \\ a \parr Q_1 \end{array} \qquad \text{and} \qquad \Pi_2 = \begin{array}{c} \Pi'_2 \Big\| \\ \mathsf{ai}{\downarrow}\dfrac{J_2\{1\}}{J_2\{a \parr \bar{a}\}} \\ \Psi_2 \Big\| \\ \bar{a} \parr Q_2 \end{array} \quad .
$$

Finally then, there exists a proof in $\mathsf{SMLS}^{\downarrow}$:

$$
\begin{array}{c}
H\{1\} \\
\Pi'_1 \Big\| \\
H\{J_1\{1\}\} \\
\Pi'_2 \Big\| \\
\mathsf{ai}{\downarrow}\dfrac{H\{J_1\{J_2\{1\}\}\}}{H\{J_1\{J_2\{a \parr \bar{a}\}\}\}} \\
\{=,\mathsf{s}\} \Big\| \\
H\{J_1\{\bar{a} \parr J_2\{a\}\}\} \\
\{=,\mathsf{s}\} \Big\| \\
=\dfrac{H\{J_1\{\bar{a}\} \parr J_2\{a\}\}}{H\{J_1\{\bot \parr \bar{a}\} \parr J_2\{a \parr \bot\}\}} \\
\Psi_1 \Big\| \\
H\{Q_1 \parr J_2\{a \parr \bot\}\} \\
\Psi_2 \Big\| \\
H\{Q_1 \parr Q_2\} \\
\Phi \Big\| \\
H\{K\} \\
\chi \Big\| \\
S\{\bot\}
\end{array} \quad ,
$$

where the Lemma of Exercise 3.5.3 has been applied twice.

□

We can apply this result to every cut starting from the topmost, to eliminate all of them.

Exercise 4.2.7. Prove shallow splitting, context reduction and cut-elimination for BV.

The shallow splitting and context reduction results tell us in what way we can split a proof to get independent subroofs. The rearrengement of these subproofs can not only be used to show cut-elimination, but also more generally toshow the admissibility of all the up-rules.

Exercise 4.2.8. Use shallow splitting and context reduction to prove that the rule $\alpha\uparrow$ is admissible.

4.3 Notes

Decomposition in the calculus of structures has first been observed for MELL in [GS01]. The first proof has been given by Straßburger in in [Str03b], and a slightly simplified version can be found in Straßburger's PhD thesis [Str03a]. Similar decomposition results exist for many systems, the local version of SKS [BT01, Gun09], the local system SLLS for linear logic [Str02a, Str03a], SBV [GS01], SNEL [GS02, SG11], and a system for intuitionistic logic [GS14].

The splitting methodology is due to Alessio Guglielmi and was first introduced in [Gug07]. Cut-elimination via splitting has been shown to work in a vast array of deep inference systems: full propositional linear logic [Str03a, CGS11], the mixed commutative/non-commutative logic BV [Gug07], its extension with linear exponentials NEL [GS02, GS11], and MAV1 [HTAC19] (which is BV extended with additives and quantifiers), and finally also linear logic extended with sub-additives [Hor19].

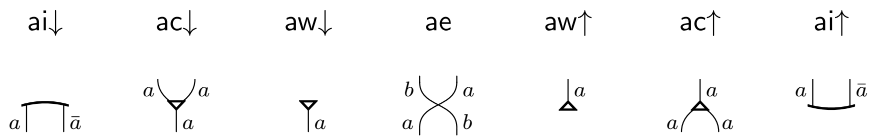


Figure 22: Generators for atomic flows

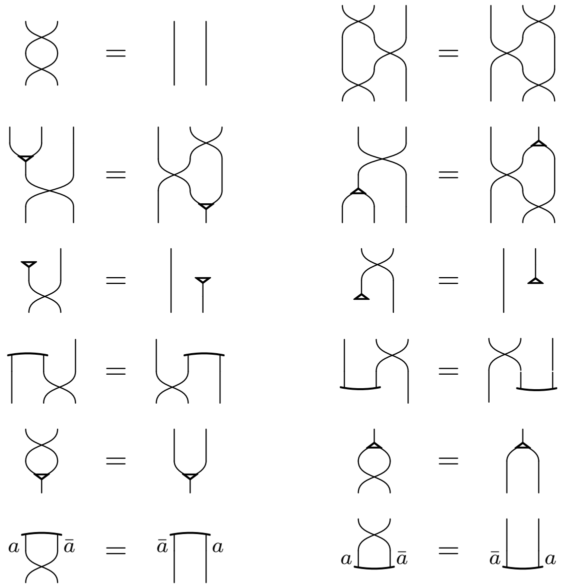


Figure 23: Relations for atomic flows

5 Atomic Flows

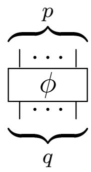
In the the first section of these lecture notes, we have seen several *syntactic* formalisms to denote proofs. In this section, we begin to remove syntax. The main motivation is to understand what makes a proof. What are the essential ingredients? What is the *essence* of a proof?

The idea behind *atomic flows*, that we study in this section, is to forget about the connectives that are used to compose the formulas, and only look at the atoms, and how the atoms are created, deleted, and moved around in a proof.

The technical motivation for atomic flows is that they give us an elegant new method of proving cut elimination in the calculus of structures; and additionally we get some new insight to answer the question posed in Exercise 4.1.4.

5.1 Basic definitions and properties

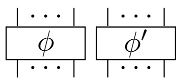
We start from a countable set $\mathscr{A}$ of *atomic types*, equipped with an involutive bijection $\bar{(\cdot)} : \mathscr{A} \to \mathscr{A}$, such that for all $a \in \mathscr{A}$, we have $\bar{a} \neq a$ and $\bar{\bar{a}} = a$. A *(flow) type* is a finite list of atomic types, denoted by $p, q, r, \ldots$, and we write $p \mid q$ for the list concatenation of p and q, and we write 0 for the empty list. An *atomic flow* $\phi : p \to q$ is a two-dimensional diagram, written as



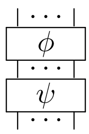
where p is the *input type* and q is the *output type*. The number of edges corresponds to the lengths of the lists, and each edge is labelled by the corresponding list element. For each type q, we have the *identity flow* id_q:

$$|\cdots|$$

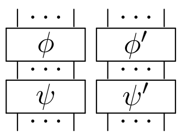
We can compose atomic flows horizontally: for $\phi\colon p \to q$ and $\phi'\colon p' \to q'$, we get $\phi|\phi'\colon p|p' \to q \mid q'$ of the shape



And we can compose atomic flows vertically: for $\phi\colon p \to q$ and $\psi\colon q \to r$, we get $\psi\circ\phi\colon p \to r$ of the shape



For $\phi\colon p \to q$ we have $\phi \circ \mathsf{id}_p = \phi = \mathsf{id}_q \circ \phi$ and $\phi \mid \mathsf{id}_0 = \phi = \mathsf{id}_0 \mid \phi$. We also have $(\psi \circ \phi) \mid (\psi' \circ \phi') = (\psi \mid \psi') \circ (\phi \mid \phi')$ which is pictured as



Finally, we have to give a set of generators and relations, which is done in Figures 22 and 23.

The generators in Figure 22 are called $\mathsf{ai}{\downarrow}$ (*atomic interaction down*), $\mathsf{ac}{\downarrow}$ (*atomic contraction down*), $\mathsf{aw}{\downarrow}$ (*atomic weakening down*), ae (*atomic exchange*), $\mathsf{aw}{\uparrow}$ (*atomic weakening up*), $\mathsf{ac}{\uparrow}$ (*atomic contraction up*), and $\mathsf{ai}{\uparrow}$ (*atomic interaction up*). The typing information in Figure 22 says that

- for $\mathsf{ai}{\downarrow}$ (resp. $\mathsf{ai}{\uparrow}$) the two output edges (resp. input edges) carry opposite atomic types,
- for $\mathsf{ac}{\downarrow}$ (resp. $\mathsf{ac}{\uparrow}$) all input and output edges carry the same atomic type,
- for $\mathsf{aw}{\downarrow}$ (resp. $\mathsf{aw}{\uparrow}$) there are no typing restrictions, and
- for ae, the left input has to carry the same type as the right output, and vice versa.

We will see in the next section that it is no coincidence that these generators carry the same names as the atomic inference rules in SKS.

When picturing an atomic flow we will omit the typing when this information is irrelevant or clear from context, as done in Figure 23. The typing is needed for two reasons: first, we need to exclude illegal flows like

 .

Note, that for this it would suffice to have only two types, $+$ and $-$. However, the second reason for having the types here is the use of the atomic flows as tool for proof transformations.

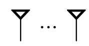
Definition 5.1.1. For a given atomic flow diagram ϕ, we define its *atomic flow graph* G_ϕ to be the directed acyclic graph whose vertices are the generators $\mathsf{ai}{\downarrow}$, $\mathsf{ai}{\uparrow}$, $\mathsf{ac}{\downarrow}$, $\mathsf{ac}{\uparrow}$, $\mathsf{aw}{\downarrow}$, $\mathsf{aw}{\uparrow}$ (i.e., all except ae) appearing in ϕ, whose incoming (resp. outcoming) edges are the incoming (resp. outcoming) edges of ϕ, and whose inner edges are downwards oriented as indicated by the flow diagram for ϕ. A *path* in ϕ is a path in G_ϕ.

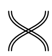
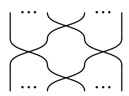
Remark 5.1.2. If we label the edges in G_ϕ by the corresponding atomic type, then for every path in ϕ, all its edges carry the same label.

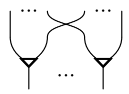
Notation 5.1.3. For making large atomic flows easier to read, we introduce the following notation:

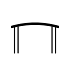
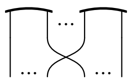
$$\| \quad \text{abbreviates} \quad | \cdots |$$

This can be extended to all generators:

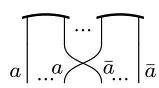
 abbreviates 

 abbreviates 

 abbreviates 

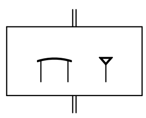
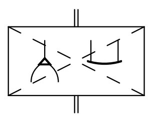
 abbreviates 

And similarly for $\mathsf{aw}{\uparrow}$, $\mathsf{ac}{\uparrow}$, and $\mathsf{ai}{\uparrow}$. In each case we allow the number of edges to be 0, which then yields the empty flow. Moreover, if we label an abbreviation with atomic type a, we mean that each edge being abbreviated has type a. For instance:

abbreviates


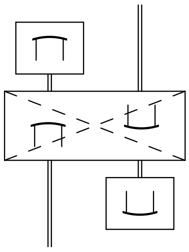
.

Notation 5.1.4. A box containing some generators stands for an atomic flow generated only from these generators, and a box containing some generators crossed out stands for an atomic flow that does not contain any of these generators. For example, the two diagrams

 and 

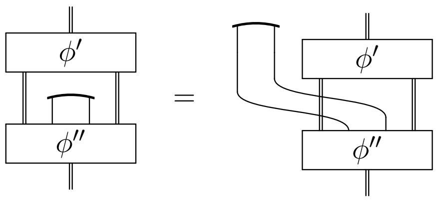
stand for a flow that contains only $\mathsf{ai}{\downarrow}$ and $\mathsf{aw}{\downarrow}$ generators and a flow that does not contain any $\mathsf{ac}{\uparrow}$ and $\mathsf{ai}{\uparrow}$ generators, respectively.

Proposition 5.1.5. *Every atomic flow ϕ can be written in the following form:*



(29)

Proof. Let ϕ be given and pick an arbitrary occurrence of $\mathsf{ai}{\downarrow}$ inside ϕ. Then ϕ can be written as shown on the left below.

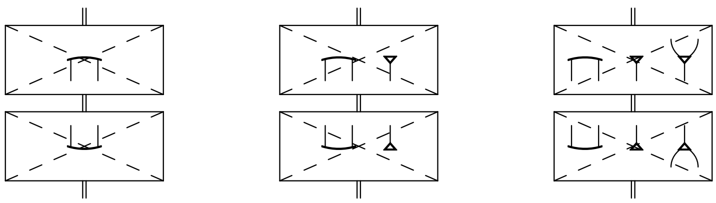


(30)

The equality follows by induction on the number of vertical edges to cross, For $\mathsf{ai}{\uparrow}$ we proceed dually. □

Exercise 5.1.6. Compare this proof to the construction in (20) and to the proof of Theorem 4.1.1.

Definition 5.1.7. An atomic flow is *weakly streamlined* (resp., *streamlined* and *strongly streamlined*) if it can be represented as the flow on the left (resp., in the middle and on the right):



. (31)

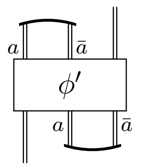
Proposition 5.1.8. *An atomic flow ϕ is weakly streamlined if and only if in G_ϕ there is no path from an $\mathsf{ai}{\downarrow}$-vertex to an $\mathsf{ai}{\uparrow}$-vertex.*

Proof. Immediate from (30), read from right to left. □

Definition 5.1.9. An atomic flow ϕ is *weakly streamlined with respect to an atomic type a* if in G_ϕ there is no path from an $\mathsf{ai}{\downarrow}$-vertex to an $\mathsf{ai}{\uparrow}$-vertex, whose edges are labelled by a or $\bar{a}$.

Definition 5.1.10. Let a be an atomic type. An atomic flow ϕ is ai*-free with respect to a* if ϕ does not contain any $\mathsf{ai}{\downarrow}$ generators whose outputs are typed by a and $\bar{a}$, and ϕ does not contain any $\mathsf{ai}{\uparrow}$ generators whose inputs are typed by a and $\bar{a}$.

Proposition 5.1.11. *Let a be an atomic type. Then every atomic flow ϕ can be written as*

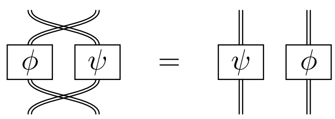


$$, \tag{32}$$

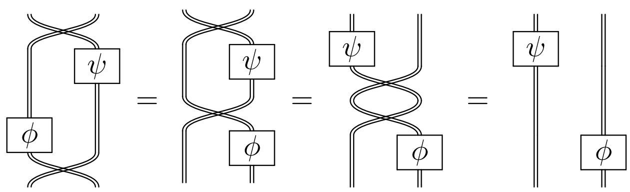
where ϕ' is ai*-free with respect to a.*

Proof. Repeatedly apply the construction of the proof of Proposition 5.1.5 (and the relations in the last line of Figure 23). ☐

Proposition 5.1.12. *For any two atomic flows ϕ and ψ, we have*



Proof. We have

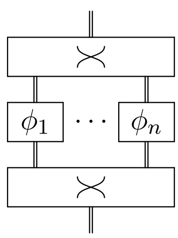


Where the first equality follows by induction on the size of ϕ, the second by induction on the size of ψ, and the third from repeatedly applying the equations in the first line of Figure 23. ☐

Definition 5.1.13. An atomic flow ϕ is called *pure* if all edges carry the same atomic type. It is called *semi-pure* if only two different atomic types a and b occur in ϕ with $b = \bar{a}$.

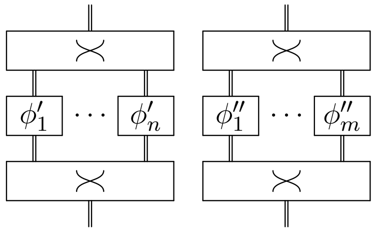
Note that a pure atomic flow cannot contain any ai↓ nor ai↑ generators.

Proposition 5.1.14. *Every atomic flow can be written as*

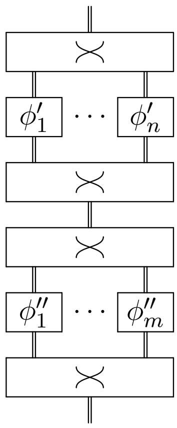


where ϕ_1, ..., ϕ_n are all semi-pure.

Proof. We proceed by induction on the size of ϕ. If ϕ is a generator or id_a for some atomic type a, then the result is trivial. If $\phi = \phi' \mid \phi''$, then by induction hypothesis we have that ϕ is equal to

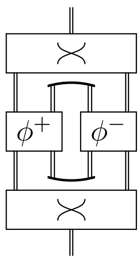


and the result follows from Proposition 5.1.12. If $\phi = \phi' \circ \phi''$, then by induction hypothesis we have that ϕ is equal to



Because of Proposition 5.1.12, we can assume that the edges in ϕ'_i and ϕ''_i carry the same atomic types, and by allowing the empty flow, we can assume that $n = m$. Then, the two exchange boxes in the middle must compose to the identity. $\square$

Proposition 5.1.15. *Every semi-pure atomic flow ϕ can be written as*



where ϕ^+ and ϕ^- are both pure.

Proof. First apply Proposition 5.1.11 to get a flow of shape (32). Then apply the construction of the previous proof to ϕ'. $\square$

$$
\begin{array}{r c}
 & (a \wedge (\bar{a} \vee \mathsf{t})) \wedge \bar{a} \\ \hline
\mathsf{ai}\!\downarrow & (a \wedge (\bar{a} \vee (\bar{a} \vee a))) \wedge \bar{a} \\ \hline
= & (a \wedge ((\bar{a} \vee \bar{a}) \vee a)) \wedge \bar{a} \\ \hline
\mathsf{s} & ((a \wedge (\bar{a} \vee \bar{a})) \vee a) \wedge \bar{a} \\ \hline
\mathsf{ac}\!\downarrow & ((a \wedge \bar{a}) \vee a) \wedge \bar{a} \\ \hline
\mathsf{ai}\!\uparrow & (\mathsf{f} \vee a) \wedge \bar{a} \\ \hline
= & a \wedge \bar{a} \\ \hline
\mathsf{ac}\!\uparrow & (a \wedge a) \wedge \bar{a} \\ \hline
= & a \wedge (a \wedge \bar{a}) \\ \hline
\mathsf{ai}\!\uparrow & a \wedge \mathsf{f}
\end{array}
\qquad
= \dfrac{\left(\mathsf{s}\,\dfrac{a \wedge \left(\bar{a} \vee \mathsf{ai}\!\downarrow \dfrac{\mathsf{t}}{\bar{a} \vee a}\right)}{\mathsf{ai}\!\uparrow \dfrac{a \wedge \mathsf{ac}\!\downarrow \dfrac{\bar{a} \vee \bar{a}}{\bar{a}}}{\mathsf{f}} \vee \mathsf{ac}\!\uparrow \dfrac{a}{a \wedge a}} \wedge \bar{a} \right)}{a \wedge \mathsf{ai}\!\uparrow \dfrac{a \wedge \bar{a}}{\mathsf{f}}}
$$

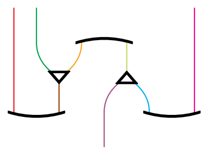
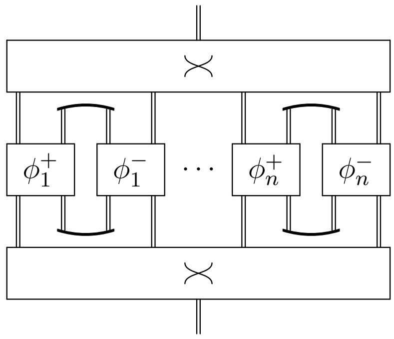


Figure 24: Example of translating a derivation in deep inference to atomic flows

Theorem 5.1.16. *Every atomic flow can be written as*



where ϕ_1^+, ϕ_1^-, ..., ϕ_n^+, ϕ_n^- *are all pure.*

Proof. Immediate from Propositions 5.1.14 and 5.1.15. □

5.2 From formal derivations to atomic flows

In this section we show how formal derivation are translated into atomic flows. Let us emphasize that this is not restricted to deep inference formalisms. We show it here for the sequent calculus, for the calculus of structures, and for open deduction.

We can assign to each formula, sequent, or list of sequents its flow type by forgetting the structural information of $\wedge$, $\vee$, t and f, and simply keeping the list of atomic types as they occur in the formulas. For a formula A, we denote this type by $\mathit{fl}(A)$.

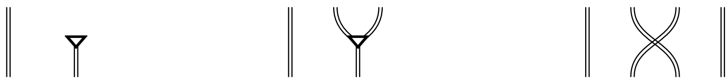
Now we can assign atomic flows to inference rules. We begin with the system shown in Figure 5. The two rules

$$
\wedge\,\frac{\vdash \Gamma, A \qquad \vdash B, \Delta}{\vdash \Gamma, A \wedge B, \Delta} \qquad \text{and} \qquad \vee\,\frac{\vdash \Gamma, A, B}{\vdash \Gamma, A \vee B}
$$

are translated into the identity flows $\mathsf{id}_{\mathit{fl}(\Gamma)} \mid \mathsf{id}_{\mathit{fl}(A)} \mid \mathsf{id}_{\mathit{fl}(B)} \mid \mathsf{id}_{\mathit{fl}(\Delta)}$ and $\mathsf{id}_{\mathit{fl}(\Gamma)} \mid \mathsf{id}_{\mathit{fl}(A)} \mid \mathsf{id}_{\mathit{fl}(B)}$, respectively. The structural rules

$$\mathsf{weak}\,\frac{\vdash \Gamma}{\vdash \Gamma, A} \qquad\qquad \mathsf{cont}\,\frac{\vdash \Gamma, A, A}{\vdash \Gamma, A} \qquad\qquad \mathsf{exch}\,\frac{\vdash \Delta, B, A, \Gamma}{\vdash \Delta, A, B, \Gamma}$$

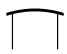
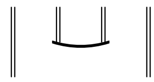
are translated into the flows

 ,

respectively. Finally, the (atomic) identity and cut rules

$$\mathsf{id}\,\frac{}{\vdash a, \bar{a}} \qquad \text{and} \qquad \mathsf{cut}\,\frac{\vdash \Gamma, A \qquad \vdash \bar{A}, \Delta}{\vdash \Gamma, \Delta}$$

are translated into

 and  ,

respectively. Then, sequent derivations are translated into atomic flows by composing the translations of the rule instances. It should be obvious that proof in any sequent system can be translated into atomic flows in this manner.

However, atomic flows carry more symmetries than present in the sequent calculus. In order to be able to mirror the richness of atomic flows inside a sound and complete deductive system for classical logic, we now look at deep inference systems.

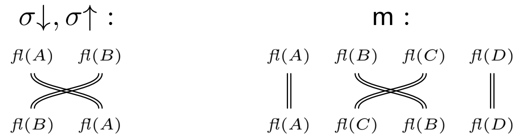
Let us look a the system SKS in the calculus of structures, shown in Figure 11. Recall that these rules can, like rewrite rules, be applied inside arbitrary contexts. For example,

$$\mathsf{ai}\!\uparrow\,\frac{(a \vee (c \wedge ((b \wedge \bar{b}) \vee \bar{a}))) \vee (b \wedge \bar{c})}{(a \vee (c \wedge (\mathsf{f} \vee \bar{a}))) \vee (b \wedge \bar{c})}$$

is a correct application of the rule $\mathsf{ai}\!\uparrow$ inside the context $(a \vee (c \wedge (\{\ \} \vee \bar{a}))) \vee (b \wedge \bar{c})$. A *derivation* $\Phi \colon A \to B$ in SKS is a rewrite path from A to B using the rules in Figure 11. We call A the *premise* and B the *conclusion* of Φ. A derivation is also denoted as

$$\begin{array}{c} A \\ \Phi \Big\Vert \mathscr{S} \\ B \end{array} \quad .$$

where $\mathscr{S}$ is the set of inference rules used in Φ. A *proof* of a formula A in SKS is a derivation $\Pi \colon \mathsf{t} \to A$. A proof in SKS is *cut-free* if it does not contain any instances of the rules $\mathsf{ai}\!\uparrow$, $\mathsf{aw}\!\uparrow$, or $\mathsf{ac}\!\uparrow$. Since the rules for weakening, contraction, and identity and cut are already in atomic form in SKS, it is straightforward to translate SKS-derivations into atomic flows. Formally, we assign to each context $S\{\ \}$ a left type and a right type denoted by $l(S\{\ \})$ and $r(S\{\ \})$, containing the lists of atomic types appearing in $S\{\ \}$ on the left, respectively on the right of the hole $\{\ \}$. For example, for $S\{\ \} = (a \vee (c \wedge (\{\ \} \vee \bar{a}))) \vee (b \wedge \bar{c})$ we have $l(S\{\ \}) = \langle a, c \rangle$ and $r(S\{\ \}) = \langle \bar{a}, b, \bar{c} \rangle$. Then, for each rule r of SKS we define the *rule flow* $\mathit{fl}(\mathsf{r})$ as follows: we map the rules $\mathsf{ai}\!\downarrow$, $\mathsf{ai}\!\uparrow$, $\mathsf{ac}\!\downarrow$, $\mathsf{ac}\!\uparrow$, $\mathsf{aw}\!\downarrow$, and $\mathsf{aw}\!\uparrow$ to the corresponding generator (with the appropriate typing), and we map the rules $\sigma\downarrow$, $\sigma\uparrow$, and m to the permutation flows shown below:



All remaining rules are mapped to the identity flow. Then an inference step

$$\mathsf{r}\,\frac{S\{A\}}{S\{B\}} \quad \text{is mapped to} \quad \mathsf{id}_{l(S\{\ \})} \mid \mathit{fl}(\mathsf{r}) \mid \mathsf{id}_{r(S\{\ \})}$$

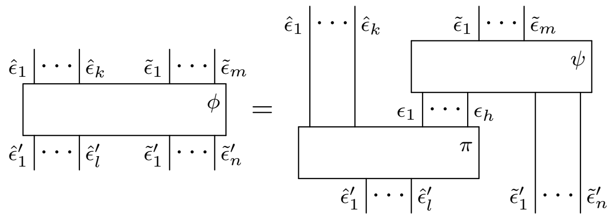
A derivation Φ is mapped to the atomic flow $\phi = \mathit{fl}(\Phi)$, which is the vertical composition of the atomic flows obtained from the inference steps in Φ.

Translating derivations in open deduction into atomic flows is even simpler, as the horizontal composition of derivation can be directly translated as horizontal composition of atomic flows. Figure 24 shows an example of a derivation in the calculus of structure, in open deduction, and its atomic flow.

Exercise 5.2.1. Define the translation from open deduction derivations to atomic flows formally.

Theorem 5.2.2. *For every flow $\phi\colon p \to q$ there is a derivation $\Phi\colon A \to B$ with $\mathit{fl}(A) = p$ and $\mathit{fl}(B) = q$ and $\mathit{fl}(\Phi) = \phi$.*

Proof. First observe that if ϕ has only ae generators, then the theorem holds trivially. Now we proceed induction on the number z_ϕ of (non-ae) generators in a given atomic flow ϕ. The cases where $z = 0$ or $z = 1$ are trivial. If $z > 1$ then ϕ can be considered as composed of two flows ψ and π, each with $z_\psi < z_\phi$ and $z_\pi < z_\phi$, as follows:



,

where $h, k, l, m, n \geq 0$ (this can possibly be done in many different ways). By the inductive hypothesis, there exist derivations $\begin{array}{c} A' \\ \Psi\Big\| \\ S\{a_1^{\epsilon_1}\}\cdots\{a_h^{\epsilon_h}\} \end{array}$ and $\begin{array}{c} T\{a_1^{\epsilon_1}\}\cdots\{a_h^{\epsilon_h}\} \\ \Pi\Big\| \\ B' \end{array}$ whose flows are, respectively, ψ and π. Using these, we can build

$$\begin{array}{c}
(T\{\mathsf{t}\}\cdots\{\mathsf{t}\} \wedge \gamma) \vee \mathsf{t} \\
{\scriptstyle (T\{\mathsf{t}\}\cdots\{\mathsf{t}\}\wedge\Psi)\vee\mathsf{t}}\Big\| \\
(T\{\mathsf{t}\}\cdots\{\mathsf{t}\} \wedge S\{a_1^{\epsilon_1}\}\cdots\{a_h^{\epsilon_h}\}) \vee \mathsf{t} \\
\Xi\Big\| \\
(T\{a_1^{\epsilon_1}\}\cdots\{a_h^{\epsilon_h}\} \wedge S\{\mathsf{f}\}\cdots\{\mathsf{f}\}) \vee \mathsf{t} \\
{\scriptstyle (\Pi\wedge\zeta\{\mathsf{f}\}\cdots\{\mathsf{f}\})\vee\mathsf{t}}\Big\| \\
(B' \wedge S\{\mathsf{f}\}\cdots\{\mathsf{f}\}) \vee \mathsf{t}
\end{array} ,$$

whose flow is ϕ, where Ξ is obtained from (24) in Section 3.5. $\square$

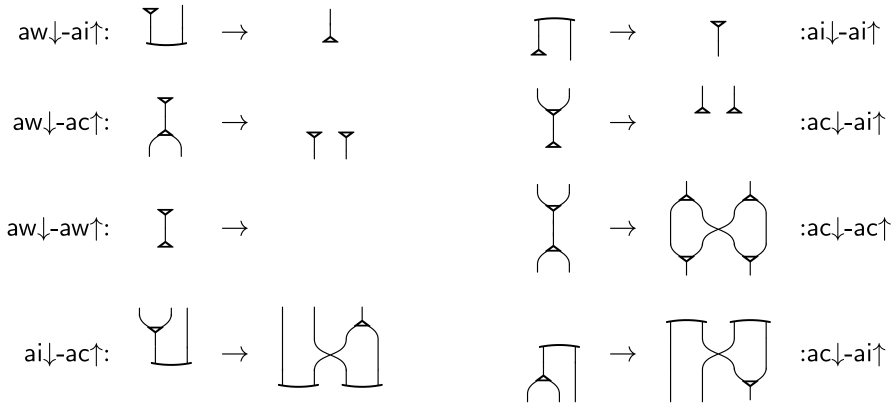


Figure 25: Local rewrite rules for atomic flows

Theorem 5.2.2 only works because the flows forget the structural information about $\wedge$, $\vee$, t and f. If we fix $\phi\colon p \to q$ together with A and B with $\mathit{fl}(A) = p$ and $\mathit{fl}(B) = q$, we can in general not provide a derivation $\Phi\colon A \to B$ with $\mathit{fl}(\Phi) = \phi$. We are thus interested in properties of atomic flows that can be lifted to derivations, in the following sense:

Definition 5.2.3. We say that a binary relation R on atomic flows *can be lifted to* SKS, if $R(\phi, \phi')$ implies that for every derivation $\Phi\colon A \to B$ with $\mathit{fl}(\Phi) = \phi$ there is a derivation $\Phi'\colon A \to B$ with $\mathit{fl}(\Phi') = \phi'$.

Remark 5.2.4. The construction in the proof of Proposition 5.1.5 does not change the flow, so there is nothing "to lift". Nonetheless, the construction in (20), which changes the derivation but does not change the underlying atomic flow, can be seen as "the lifting" of (30).

Definition 5.2.5. A derivation $\Phi\colon A \to B$ is *weakly streamlined (resp. streamlined, resp. strongly streamlined)* if $\mathit{fl}(\Phi)$ is weakly streamlined (resp. streamlined, resp. strongly streamlined).

The property *strongly streamlined* can indeed be seen as the up-down symmetric generalization being *cut-free*:

Proposition 5.2.6. *Every strongly streamlined proof in* SKS *is cut-free.*

Proof. If the premise of a strongly streamlined derivation is t, then the upper box of its flow, as given on the right in (31), must be empty. $\square$

5.3 Local Flow Transformations

We denote by $\stackrel{\mathsf{cw}}{\to}$ the rewrite relation on atomic flows generated by the rules shown in Figure 25. There are two important immediate observations about this relation: first, it can be lifted to SKS and second, it is locally confluent.

Theorem 5.3.1. *The relation* $\stackrel{\mathsf{cw}}{\to}$ *can be lifted to* SKS.

$$
\begin{array}{rc}
 & T\{\mathsf{f}\} \\ \cline{2-2}
\mathsf{aw}\!\downarrow & T\{a\} \\
\Phi & \Vert \\
 & S\{a \wedge \bar{a}\} \\ \cline{2-2}
\mathsf{ai}\!\uparrow & S\{\mathsf{f}\}
\end{array}
\quad \xrightarrow{\mathsf{aw}\downarrow\text{-}\mathsf{ai}\uparrow} \quad
\begin{array}{rc}
 & T\{\mathsf{f}\} \\
\Phi\{a/\mathsf{f}\} & \Vert \\
 & S\{\mathsf{f} \wedge \bar{a}\} \\ \cline{2-2}
\mathsf{aw}\!\uparrow & S\{\mathsf{f} \wedge \mathsf{t}\} \\ \cline{2-2}
= & S\{\mathsf{f}\}
\end{array}
\qquad\qquad
\begin{array}{rc}
 & T\{\mathsf{f}\} \\ \cline{2-2}
\mathsf{aw}\!\downarrow & T\{a\} \\
\Phi & \Vert \\
 & S\{a\} \\ \cline{2-2}
\mathsf{aw}\!\uparrow & S\{\mathsf{t}\}
\end{array}
\quad \xrightarrow{\mathsf{aw}\downarrow\text{-}\mathsf{aw}\uparrow} \quad
\begin{array}{rc}
 & T\{\mathsf{f}\} \\
\Phi\{a/\mathsf{f}\} & \Vert \\
 & S\{\mathsf{f}\} \\ \cline{2-2}
= & S\{(\mathsf{t} \wedge \mathsf{f}) \vee (\mathsf{f} \wedge \mathsf{t})\} \\ \cline{2-2}
\mathsf{m} & S\{(\mathsf{t} \vee \mathsf{f}) \wedge (\mathsf{f} \vee \mathsf{t})\} \\ \cline{2-2}
= & S\{\mathsf{t}\}
\end{array}
$$

$$
\begin{array}{rc}
 & T\{\mathsf{f}\} \\ \cline{2-2}
\mathsf{aw}\!\downarrow & T\{a\} \\
\Phi & \Vert \\
 & S\{a\} \\ \cline{2-2}
\mathsf{ac}\!\uparrow & S\{a \wedge a\}
\end{array}
\quad \xrightarrow{\mathsf{aw}\downarrow\text{-}\mathsf{ac}\uparrow} \quad
\begin{array}{rc}
 & T\{\mathsf{f}\} \\
\Phi\{a/\mathsf{f}\} & \Vert \\
 & S\{\mathsf{f}\} \\ \cline{2-2}
\mathsf{nm}\!\downarrow & S\{\mathsf{f} \wedge \mathsf{f}\} \\ \cline{2-2}
\mathsf{aw}\!\downarrow & S\{a \wedge \mathsf{f}\} \\ \cline{2-2}
\mathsf{aw}\!\downarrow & S\{a \wedge a\}
\end{array}
\qquad\qquad
\begin{array}{rc}
 & T\{a \vee a\} \\ \cline{2-2}
\mathsf{ac}\!\downarrow & T\{a\} \\
\Phi & \Vert \\
 & S\{a \wedge \bar{a}\} \\ \cline{2-2}
\mathsf{ai}\!\uparrow & S\{\mathsf{f}\}
\end{array}
\quad \xrightarrow{\mathsf{ac}\downarrow\text{-}\mathsf{ai}\uparrow} \quad
\begin{array}{rc}
 & T\{a \vee a\} \\
\Phi\{a/a\vee a\} & \Vert \\
 & S\{(a \vee a) \wedge \bar{a}\} \\ \cline{2-2}
\mathsf{ac}\!\uparrow & S\{(a \vee a) \wedge (\bar{a} \wedge \bar{a})\} \\ \cline{2-2}
=,\mathsf{s} & S\{(a \vee (a \wedge \bar{a})) \wedge \bar{a}\} \\ \cline{2-2}
=,\mathsf{s} & S\{(a \wedge \bar{a}) \vee (a \wedge \bar{a})\} \\ \cline{2-2}
\mathsf{ai}\!\uparrow & S\{\mathsf{f} \vee (a \wedge \bar{a})\} \\ \cline{2-2}
= & S\{a \wedge \bar{a}\} \\ \cline{2-2}
\mathsf{ai}\!\uparrow & S\{\mathsf{f}\}
\end{array}
$$

Figure 26: Lifting of the rules in the left column in Figure 25

Proof. Let ψ be an atomic flow with redex , and let ψ' be the result of applying the rule $\mathsf{aw}\!\downarrow$-$\mathsf{ai}\!\uparrow$ to it, i.e., rewriting that redex with . Then any derivation Ψ having ψ as atomic flow must be of shape as indicated in the upper left of Figure 26. Let Ψ' be the derivation obtained from Ψ by performing the transformation indicated Figure 26. There $\Phi\{a/\mathsf{f}\}$ stands for the derivation Φ in which in every line the unique occurrence of the a that corresponds to the path between the $\mathsf{aw}\!\downarrow$ and the $\mathsf{ai}\!\uparrow$ is replaced by f. Then the atomic flow of Ψ' is ψ'. For the other rules we can proceed similarly. Figure 26 shows the corresponding derivations for the rewrite rules of $\overset{\mathsf{cw}}{\to}$ shown on the left in Figure 25. We leave the the other four rules as an exercise. $\square$

Exercise 5.3.2. Complete the proof for Theorem 5.3.1 by showing the reamining four cases.

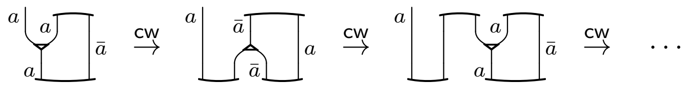
Proposition 5.3.3. *The rewrite relation* $\overset{\mathsf{cw}}{\to}$ *is locally confluent.*

Proof. The result follows from a case analysis on the critical peaks, which are:

and their duals. $\square$

Exercise 5.3.4. Fill in the details of this proof.

However, in general the reduction $\overset{\mathsf{cw}}{\to}$ is not terminating. This can easily be seen by the following example:



The reason is that there can be cycles composed of paths connecting instances of the $\mathsf{ai}{\downarrow}$ and $\mathsf{ai}{\uparrow}$ generators. The purpose of the notion "weakly streamlined" (Definition 5.1.7) is precisely to avoid such a situation.

Theorem 5.3.5. *Every weakly streamlined atomic flow has a unique normal form with respect to $\overset{\mathsf{cw}}{\to}$, and this normal form is strongly streamlined.*

Proof. First, we have to show the existence of a normal form, i.e., termination of the rewrite system. For this, observe that the generators $\mathsf{ac}{\uparrow}$ and $\mathsf{aw}{\uparrow}$ move upwards in the flow and the generators $\mathsf{ac}{\downarrow}$ and $\mathsf{aw}{\downarrow}$ move down. For each $\mathsf{ac}{\uparrow}$- and $\mathsf{aw}{\uparrow}$-generator x, we let x_{cw} be the number of $\mathsf{ac}{\downarrow}$ and $\mathsf{aw}{\downarrow}$-generators above it, and x_i be the number of $\mathsf{ai}{\downarrow}$ above it. Dually we define x_{cw} and x_i for $\mathsf{ac}{\downarrow}$- and $\mathsf{aw}{\downarrow}$-generators x. Note that x_i is either 1 or 0; and for an $\mathsf{ac}{\downarrow}$ (resp. $\mathsf{ac}{\uparrow}$) created bu the rewrite $\mathsf{ac}{\downarrow}$-$\mathsf{ai}{\uparrow}$ (resp. $\mathsf{ai}{\downarrow}$-$\mathsf{ac}{\uparrow}$), this x_i always 0 (because by Proposition 5.1.8 there is no path between an $\mathsf{ai}{\downarrow}$ and an $\mathsf{ai}{\uparrow}$). Now, for each generator x of type $\mathsf{ac}{\uparrow}$, $\mathsf{aw}{\uparrow}$, $\mathsf{ac}{\downarrow}$, and $\mathsf{aw}{\downarrow}$ we define its *value* to be the pair $\langle x_i, x_{cw} \rangle$ ordered lexicographically. Now observe that each rewrite step either removes generators or replaces them by other generators of smaller value. Furthermore, the generators in the atomic flow that are not touched by a rewrite step do not change their value. Thus, we can define the *value* of an atomic flow to be the multiset of the values of its generators (we do not count ae, $\mathsf{ai}{\downarrow}$, and $\mathsf{ai}{\uparrow}$). Then each rewrite step reduces this value according to the standard multiset ordering. Since this ordering is well-founded, the process terminates.

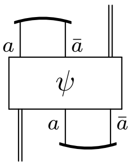
Uniqueness of the normal form follows from Proposition 5.3.3 (because local confluence and termination entail confluence). Since $\overset{\mathsf{cw}}{\to}$ preserves the property of being weakly streamlined, and in the normal form there are no more redexes for $\overset{\mathsf{cw}}{\to}$, there is no generator $\mathsf{ai}{\downarrow}$, $\mathsf{aw}{\downarrow}$, $\mathsf{ac}{\downarrow}$ above a generator $\mathsf{ai}{\uparrow}$, $\mathsf{aw}{\uparrow}$, $\mathsf{ac}{\uparrow}$, which means that the atomic flow is strongly streamlined. □

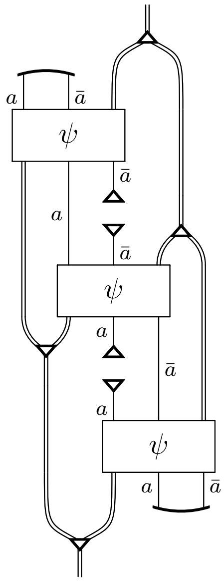
Exercise 5.3.6. Look back at your guess in Exercise 4.1.4. Do you stick to your opinion, or do you change your mind. Use what you have learned in Sections 5.2 and 5.3 to prove it.

5.4 Global Flow Transformations

In this section we show a method that can transform any atomic flow into a weakly streamlined one. The actual challenge for doing so is to find an operation that can be lifted to derivations in SKS. The key idea of the method we present here is to breaks paths in the flow without removing any edge. For this reason it is called the *path breaker*.

Definition 5.4.1. Let ϕ be an atomic flow of the shape on the left below, where the edges of the selected $\mathsf{ai}{\downarrow}$ and $\mathsf{ai}{\uparrow}$ generators carry the same atomic types, and let ϕ' be the atomic flow on the right below.

$\phi \quad =$ 

$\phi' \quad =$ 

. (33)

Then we call ϕ' a *path breaker of* ϕ *with respect to* a, and write $\phi \overset{\mathsf{pb}}{\to}_a \phi'$.

Lemma 5.4.2. *Let* ϕ *and* ϕ' *be given with* $\phi \overset{\mathsf{pb}}{\to}_a \phi'$*, and let* b *be any atomic type. If* ϕ *is weakly streamlined with respect to* b*, then so is* ϕ'*.*

Proof. The only edges connecting an output of one copy of ψ to an input of another copy of ψ are typed by a and $\bar{a}$. Thus, the lemma is evident for $b \neq a$ and $b \neq \bar{a}$. Let us now assume $b = a$ and proceed by contradiction. Assume there is an $\mathsf{ai}{\downarrow}$ generator connected to an $\mathsf{ai}{\uparrow}$ generator via a path typed by a. If this is inside a copy of ψ, we have a contradiction; if it passes through the a-edge between the upper and the middle copy of ψ in (33), then this path connects to the $\mathsf{ai}{\downarrow}$ on the left in (33), which also is a contradiction. Similarly for a path typed by $\bar{a}$. □

Lemma 5.4.3. *Let* ϕ*,* ψ*, and* a *be given as on the left in* (33)*, and let* $\phi \overset{\mathsf{pb}}{\to}_a \phi'$*. If* ψ *is* ai*-free with respect to* a*, then* ϕ' *is weakly streamlined with respect to* a*.*

Proof. For not being weakly streamlined with respect to a, we would need a path connecting the upper $\mathsf{ai}{\downarrow}$ in (33) to the lower $\mathsf{ai}{\uparrow}$. However, such a path must pass through both the evidenced edge of type a and the evidenced edge of type $\bar{a}$, which is impossible (see Remark 5.1.2). □

Lemmas 5.4.2 and 5.4.3 justify the name *path breaker* for the atomic flow on the right of (33). It breaks all paths between the upper $\mathsf{ai}{\downarrow}$ and the lower $\mathsf{ai}{\uparrow}$ in the flow on the left of (33), and it does not introduce any new paths. Furthermore, the interior of the flow ψ is not touched. And as promissed, the path breaker can be lifted to SKS.

Lemma 5.4.4. *The relation* $\overset{\mathsf{pb}}{\to}_a$ *can be lifted to* SKS*.*

Proof. Let $\Phi\colon A \to B$ with $\mathit{fl}(\Phi) = \phi$ and a be given. By applying (20) we have a derivation

$$\begin{array}{r c}
\mathsf{ai}\!\downarrow,\mathsf{t}\!\downarrow & \dfrac{A}{(a \vee \bar{a}) \wedge A} \\
 & {\scriptstyle \Psi}\Big\| \\
\mathsf{t}\!\uparrow,\mathsf{ai}\!\uparrow & \dfrac{B \vee (a \wedge \bar{a})}{B}
\end{array} \quad ,$$

with $\mathit{fl}(\Psi) = \psi$. We also have the derivations

$$\begin{array}{r c}
\mathsf{aw}\!\uparrow & \dfrac{(B \vee (a \wedge \bar{a})) \wedge A}{(B \vee (a \wedge \mathsf{t})) \wedge A} \\
= & \dfrac{}{(B \vee (a \vee \mathsf{f})) \wedge A} \\
\mathsf{aw}\!\downarrow & \dfrac{}{(B \vee (a \vee \bar{a})) \wedge A} \\
\mathsf{s} & \dfrac{}{B \vee ((a \vee \bar{a}) \wedge A)}
\end{array}
\qquad \text{and} \qquad
\begin{array}{r c}
\mathsf{aw}\!\uparrow & \dfrac{(B \vee (a \wedge \bar{a})) \wedge A}{(B \vee (\mathsf{t} \wedge \bar{a})) \wedge A} \\
= & \dfrac{}{(B \vee (\mathsf{f} \vee \bar{a})) \wedge A} \\
\mathsf{aw}\!\downarrow & \dfrac{}{(B \vee (a \vee \bar{a})) \wedge A} \\
\mathsf{s} & \dfrac{}{B \vee ((a \vee \bar{a}) \wedge A)}
\end{array}$$

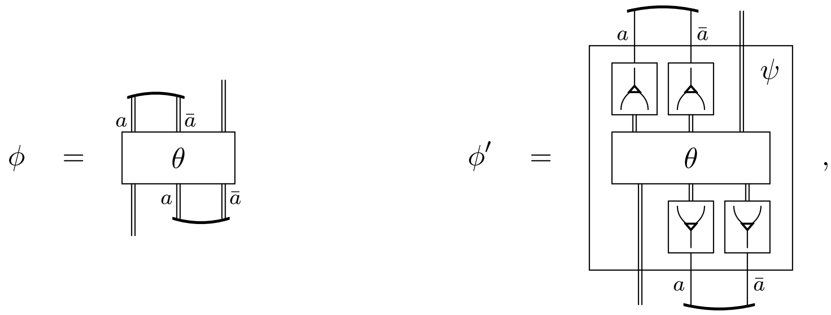
that we call Φ_a and $\Phi_{\bar{a}}$, respectively. We can now build

$$\begin{array}{c}
A \\
\Big\|{\scriptstyle \{\mathsf{c}\uparrow,\mathsf{ai}\downarrow,=\}} \\
(((a \vee \bar{a}) \wedge A) \wedge A) \wedge A \\
{\scriptstyle (\Psi \wedge A) \wedge A}\Big\| \\
((B \vee (a \wedge \bar{a})) \wedge A) \wedge A \\
{\scriptstyle \Phi_a \wedge A}\Big\| \\
(B \vee ((a \vee \bar{a}) \wedge A)) \wedge A \\
{\scriptstyle (B \vee \Psi) \wedge A}\Big\| \\
B \vee ((B \vee (a \wedge \bar{a})) \wedge A) \\
{\scriptstyle B \vee \Phi_{\bar{a}}}\Big\| \\
B \vee (B \vee ((a \vee \bar{a}) \wedge A)) \\
{\scriptstyle B \vee (B \vee \Psi)}\Big\| \\
B \vee (B \vee (B \vee (a \wedge \bar{a}))) \\
\Big\|{\scriptstyle \{\mathsf{c}\downarrow,\mathsf{ai}\uparrow,=\}} \\
B
\end{array} \quad ,$$

whose atomic flow is as shown on the right of (33). □

Exercise 5.4.5. Write the derivation in the proof of Lemma 5.4.4 using open deduction.

We now have to find a way to convert any atomic flow ϕ into one of shape on the left of (33) with ψ being ai-free with respect to a. For this, notice that by Proposition 5.1.11, we can write ϕ as shown on the left below where θ is ai-free with respect to a. This can be transformed into a flow ϕ' on the right below:



(34)

which is of the desired shape and fulfills the condition of Lemma 5.4.3.

Definition 5.4.6. Let ϕ and ϕ' of shape as in (34) be given. If θ is ai-free with respect to a, then we call ϕ' a *taming of ϕ with respect to a*, and write $\phi \overset{\mathsf{tm}}{\to}_a \phi'$.

Lemma 5.4.7. *Let ϕ and ϕ' be given with $\phi \overset{\mathsf{tm}}{\to}_a \phi'$, and let b be any atomic type. If ϕ is weakly streamlined with respect to b, then so is ϕ'.*

Proof. Immediate from (34). □

Lemma 5.4.8. *The relation $\overset{\mathsf{tm}}{\to}_a$ can be lifted to* SKS.

Proof. Let a and $\Phi\colon A \to B$ with $\mathit{fl}(\Phi) = \phi$ (as shown on the left of (34)) be given. By repeatedly applying (20) we get the derivation on the left below

$$
\begin{array}{c}
A \\
\Big\| \{\mathsf{ai}{\downarrow},=\} \\
(a \vee \bar{a}) \wedge \cdots \wedge (a \vee \bar{a}) \wedge A \\
\Theta \Big\| \\
B \vee (a \wedge \bar{a}) \vee \cdots \vee (a \wedge \bar{a}) \\
\Big\| \{\mathsf{ai}{\uparrow},=\} \\
B
\end{array}
\quad \rightsquigarrow \quad
\begin{array}{c}
\mathsf{t}{\downarrow}\,\dfrac{A}{\mathsf{t} \wedge A} \\
\mathsf{ai}{\downarrow},\mathsf{t}{\downarrow}\,\dfrac{\phantom{(a \vee \bar{a}) \wedge A}}{(a \vee \bar{a}) \wedge A} \\
\Big\| \{\mathsf{ac}{\uparrow},=\} \\
((a \wedge \cdots \wedge a) \vee (\bar{a} \wedge \cdots \wedge \bar{a})) \wedge A \\
\Big\| \{\mathsf{m},=\} \\
(a \vee \bar{a}) \wedge \cdots \wedge (a \vee \bar{a}) \wedge A \\
\Theta \Big\| \\
B \vee (a \wedge \bar{a}) \vee \cdots \vee (a \wedge \bar{a}) \\
\Big\| \{\mathsf{m},=\} \\
B \vee ((a \vee \cdots \vee a) \wedge (\bar{a} \vee \cdots \vee \bar{a})) \\
\Big\| \{\mathsf{ac}{\downarrow},=\} \\
\mathsf{ai}{\uparrow}\,\dfrac{B \vee (a \wedge \bar{a})}{B \vee \mathsf{f}} \\
\mathsf{t}{\uparrow}\,\dfrac{\phantom{B \vee \mathsf{f}}}{B}
\end{array}
\quad , \qquad (35)
$$

with $\mathit{fl}(\Theta) = \theta$, from which we can obtain a derivation on the right above, whose flow is as shown on the right of (34). □

Exercise 5.4.9. Write the two derivations in (35) using open deduction.

Definition 5.4.10. On atomic flows, we define the *path breaking relation* $\overset{\mathsf{PB}}{\to}$ as follows. We have $\phi \overset{\mathsf{PB}}{\to} \phi'$ if and only if there is a flow ϕ'' and an atomic type a, such that $\phi \overset{\mathsf{tm}}{\to}_a \phi'' \overset{\mathsf{pb}}{\to}_a \phi'$ and ϕ is not weakly streamlined with respect to a.

Theorem 5.4.11. *The relation* $\overset{\mathsf{PB}}{\to}$ *can be lifted to* SKS*.*

Proof. Immediate from Lemmas 5.4.8 and 5.4.4. □

Theorem 5.4.12. *The relation* $\overset{\mathsf{PB}}{\to}$ *is terminating, and its normal forms are weakly streamlined.*

Proof. Let ϕ be given. We proceed by induction on the number of atomic types occurring in ϕ, with respect to which ϕ is not weakly streamlined. Whenever we have $\phi \overset{\mathsf{PB}}{\to} \phi'$, this number is decreased by one for ϕ' (by Lemmas 5.4.2, 5.4.3, and 5.4.7). By the constructions in (33) and (34), there is always such a ϕ' if ϕ is not weakly streamlined. □

5.5 Normalizing Derivations via Atomic Flows

In this section we put the results of the previous two sections together and show how SKS derivations can be normalized using constructions on atomic flows. The basic idea is to transform an arbitrary SKS derivation first into a weakly streamlined one, and then into a strongly streamlined one, without changing premise and conclusion during the process. In other words we are going to show the following:

Theorem 5.5.1. *For every* SKS *derivation from A to B, there is a* SKS*-derivation from A to B that is strongly streamlined.*

Proof. For every SKS-derivation $\Phi\colon A \to B$ there exists a weakly-streamlined SKS-derivation $\Phi'\colon A \to B$ by Theorem 5.4.12 and Theorem 5.4.11; for every weakly-streamlined SKS-derivation $\Phi'\colon A \to B$ there exists a strongly streamlined SKS-derivation $\Phi''\colon A \to B$ by Theorem 5.3.5 and Theorem 5.3.1. □

From this we get immediately the cut elimination theorem for SKS:

Corollary 5.5.2. *For every* SKS*-proof of A, there is a cut-free* SKS*-proof of A.*

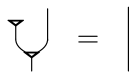
Proof. By Theorem 5.5.1 and Proposition 5.2.6. □

Exercise 5.5.3. State a version of the first decomposition theorem for the variant of SKS shown in Figure 11, and prove it using atomic flows.

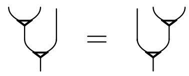
5.6 Atomic Flows as Categories

Atomic flows form a (strict) monoidal category, that we can denote by **AF**. In that category, the flow types are the objects and the atomic flows $p \to q$ are the morphisms from p to q.

We could add the relations

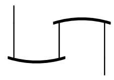


and



and their duals to the ones given in Figure 23, and this would equip every object in **AF** with a monoid and a comonoid structure. However, the results in this paper do not rely on that, and we decided to keep the set of relations minimal.

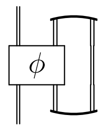
The category **AF** of atomic flows is strictly monoidal, but it is not compact closed, basically because we do not have an equality between the two atomic flows shown below:



and

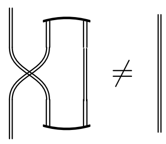
(36)

More precisely, although we can for a given atomic flow $\phi \colon p \mid x \to q \mid x$ define the atomic flow $\mathrm{Tr}^x(\phi) \colon p \to q$ as



,

the category **AF** is not traced because it does not obey yanking:



We use **SKS** to denote the category whose objects are the formulas and whose arrows are the derivations of SKS. Then the translation from SKS-derivations to atomic flows defines a forgetful functor $\mathit{fl} \colon \mathbf{SKS} \to \mathbf{AF}$. Note that this functor is independent from the fact whether the binary connectives $\wedge$ and $\vee$ are bifunctors in **SKS** (with or without monoidal structure), whether the inference rules s and m are natural transformation, and whether $\alpha\!\downarrow$, $\alpha\!\uparrow$, etc., are isomorphisms in **SKS** or not. Note that Theorem 5.2.2 does not imply that this functor is full.

5.7 Limits of Atomic Flows

We have seen that atomic flows can be very useful to get some insight into new tranformations on SKS derivations. In Theorem 5.2.2 we have seen that every atomic flow comes from some derivation. But we also have said that when we fix two formulas A and B, and have an atomic flow $\phi \colon \mathit{fl}(A) \to \mathit{fl}(B)$, then in general we do not have a SKS derivation $\Phi \colon A \to B$ with $\mathit{fl}(\Phi) = \phi$. When do we have such a derivation? Can this be decided? The answer is, of course, yes. We simply can enumerate all derivations from A to B and check if one of them has flow ϕ. The real question is:

Open Problem 5.7.1. Given two formulas A and B, and an atomic flow $\phi\colon \mathit{fl}(A) \to \mathit{fl}(B)$, can we decide in time polynomial in the size of ϕ whether there is a derivation $\Phi\colon A \to B$ with $\mathit{fl}(\Phi) = \phi$?

The answer is "probably not". Anupam Das has shown recently that there cannot be a polynomial correctness criterion for atomic flows, unless integer factoring is in **P/poly**. This leads to the question whether we can find a graphical representation for proofs which have similar properties as atomic flows, but additionally enjoy a polynomial correctness criterion. This is the motivation for Section 6.

5.8 Notes

Atomic flows have first been introduced by Guglielmi and Gundersen in [GG08] and in Gundersen's PhD thesis [Gun09] (from which we have taken the example in Figure 24). In that presentation first presentation, atomic flows have been defined as directed graphs, as done in Definition 5.1.1. Indeed, G_ϕ is the "canonical representative" of a class of flow diagrams wrt. to the equalities in Figure 23. However, with that definition the order of the input/output edges is lost, which makes the vertical composition and the mapping from formal derivations (done in Section 5.2) more difficult to define. The presentation we used here has been introduced in [GGS10]. The concept of two-dimensional diagram (that is the basis for this presentation) is due to Lafont [Laf95]. He has also shown that the generator ae together with the first two relations in Figure 23 defines the category of permutations. An important consequence is that any atomic flow $\phi : p \to p$ containing only the generator ae is equal to the identity id_p.

The local transformations defined by $\stackrel{\mathsf{cw}}{\to}$ has been studied in [GG08] where also all properties that we have shown here have first been proved. The global transformations of Section 5.4 have been introduced in [GGS10]. In [GG08] a different method for removing cycles has been used. Yet another method for eliminating cycles has recently been presented in [AGR17], and will be discussed in Section 7 of these course notes.

For more insights into the category theoretical treatment of classical proofs, with focus on medial and switch, see [Str07c] and [Lam07].

Das' result on the (probable) non-existence of a polynomial correctness criterion for atomic flows has been presented in [Das13]. More complexity related results using atomic flows can be found in [Das12, Das15].

6 Combinatorial Proofs

Combinatorial proofs have been introduced by Dominic Hughes as a way to present proofs of classical logic independent from a syntactic proof system.

6.1 Basic definitions

As before, we consider formulas (denoted by capital Latin letters $A, B, C, \ldots$) in negation normal form (NNF), generated from a countable set $\mathscr{V} = \{a, b, c, \ldots\}$ of (propositional) variables by the following grammar:

$$A, B ::= a \mid \bar{a} \mid A \wedge B \mid A \vee B \tag{37}$$

where $\bar{a}$ is the negation of a. The negation can then be defined for all formulas using the De Morgan laws:

$$\bar{\bar{A}} = A \qquad \overline{A \wedge B} = \bar{A} \vee \bar{B} \tag{38}$$

An *atom* is a variable or its negation. We use $\mathscr{A}$ to denote the set of all atoms. Sometimes we use $A \Rightarrow B$ as abbreviation for $\bar{A} \vee B$, and $A \Leftrightarrow B$ as abbreviation for $(A \Rightarrow B) \wedge (B \Rightarrow A)$. A *sequent* Γ is a multiset of formulas, written as a list separated by comma:

$$\Gamma = A_1, A_2, \ldots, A_n \tag{39}$$

We write $\bar{\Gamma}$ to denote the sequent $\bar{A}_1, \bar{A}_2, \ldots, \bar{A}_n$. We define the *size* of a sequent Γ, denoted by $\mathsf{size}(\Gamma)$, to be the number of atom occurrences in it. We write $\bigwedge\Gamma$ (resp. $\bigvee\Gamma$) for the conjunction (res. disjunction) of the formulas in Γ.

Before we can discus the notion of combinatorial proof, we need some preliminary definitions from graph theory.

Definition 6.1.1. A *(simple) graph* $\mathfrak{G} = \langle V_{\mathfrak{G}}, E_{\mathfrak{G}} \rangle$ consists of a set of *vertices* $V_{\mathfrak{G}}$ and a set of *edges* $E_{\mathfrak{G}}$ which are two-element subsets of $V_{\mathfrak{G}}$. If $E_{\mathfrak{G}}$ is not a set but a multiset, we call $\mathfrak{G}$ a *multigraph*. We omit the index $\mathfrak{G}$ when it is clear from context. For $v, w \in V$ we write vw for $\{v, w\}$. The *size* of a graph $\mathfrak{G}$, denoted by $\mathsf{size}(\mathfrak{G})$ is $\mathsf{size}(V_{\mathfrak{G}}) + \mathsf{size}(E_{\mathfrak{G}})$. A *graph homomorphism* $f\colon \mathfrak{G} \to \mathfrak{G}'$ is a function from $V_{\mathfrak{G}}$ to $V_{\mathfrak{G}'}$ such that $vw \in E_{\mathfrak{G}}$ implies $f(v)f(w) \in E_{\mathfrak{G}'}$. A simple graph $\mathfrak{G}$ is called a *cograph* if it does not contain four distinct vertices u, v, w, z with $uv, vw, wz \in E$ and $vz, zu, uw \notin E$. For a set L, a graph $\mathfrak{G}$ is *L-labeled* if every vertex of $\mathfrak{G}$ is associated with an element L, called its *label.* For two graphs $\mathfrak{G} = \langle V, E \rangle$ and $\mathfrak{G}' = \langle V', E' \rangle$, we define the operations *union* $\mathfrak{G} \vee \mathfrak{G}' = \langle V \cup V', E \cup E' \rangle$ and *join* $\mathfrak{G} \wedge \mathfrak{G}' = \langle V \cup V', E \cup E' \cup \{vv' \mid v \in V, v' \in V'\} \rangle$. If $\mathfrak{G}$ and $\mathfrak{G}'$ are L-label-led graphs, then so are $\mathfrak{G} \vee \mathfrak{G}'$ and $\mathfrak{G} \wedge \mathfrak{G}'$ where every vertex keeps its original label. For a simple graph $\mathfrak{G} = \langle V, E \rangle$, also define its *negation* $\bar{\mathfrak{G}} = \langle V, \{vw \mid v \neq w, vw \notin E\} \rangle$. If $\mathfrak{G}$ is an $\mathscr{A}$-labeled graph (where $\mathscr{A}$ is the set of atoms) then all labels are negated in $\bar{\mathfrak{G}}$. For two homomorphisms $f_1\colon \mathfrak{G}_1 \to \mathfrak{G}'_1$ and $f_2\colon \mathfrak{G}_2 \to \mathfrak{G}'_2$ such that $V_{\mathfrak{G}_1} \cap V_{\mathfrak{G}_2} = \emptyset$, we define $f_1 \vee f_2\colon \mathfrak{G}_1 \vee \mathfrak{G}_2 \to \mathfrak{G}'_1 \vee \mathfrak{G}'_2$ to be the *union* of the two homomorphisms f_1 and f_2, and $f_1 \wedge f_2\colon \mathfrak{G}_1 \wedge \mathfrak{G}_2 \to \mathfrak{G}'_1 \wedge \mathfrak{G}'_2$ to be their *join.*

Construction 6.1.2. If we associate to each atom a a single vertex labeled with a then every formula A uniquely determines a graph $\mathfrak{G}(A)$ that is constructed via the operations $\wedge$ and $\vee$. For a sequent $\Gamma = A_1, A_2, \ldots, A_n$, we define

$$\mathfrak{G}(\Gamma) = \mathfrak{G}(\bigvee\Gamma) = \mathfrak{G}(A_1) \vee \mathfrak{G}(A_2) \vee \cdots \vee \mathfrak{G}(A_n) \quad .$$

Note that this construction entails that $\overline{\mathfrak{G}(A)} = \mathfrak{G}(\bar{A})$.

Lemma 6.1.3. *For two formulas A and B, we have $\mathfrak{G}(A) = \mathfrak{G}(B)$ iff A and B are equivalent modulo associativity and commutativity of $\wedge$ and $\vee$:*

$$\begin{array}{ll} A \wedge (B \wedge C) = (A \wedge B) \wedge C & \quad A \wedge B = B \wedge A \\ A \vee (B \vee C) = (A \vee B) \vee C & \quad A \vee B = B \vee A \end{array} \tag{40}$$

Proof. Immediately from Construction 6.1.2. □

Example 6.1.4. Let $A = (a \wedge (b \vee \bar{c})) \vee (c \wedge \bar{d})$ then $\bar{A} = (\bar{a} \vee (\bar{b} \wedge c)) \wedge (\bar{c} \vee d)$. Below are the two graphs $\mathfrak{G}(A)$ and $\mathfrak{G}(\bar{A}) = \overline{\mathfrak{G}(A)}$:

(41)

Proposition 6.1.5. *A graph $\mathfrak{G}$ is a cograph iff it is constructed from a formula via Construction 6.1.2.*

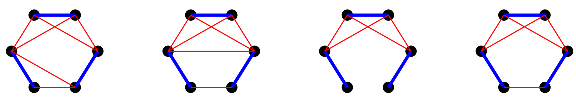
An important consequence of this and Lemma 6.1.3 is that for each cograph $\mathfrak{G}$ there is a unique (up to associativity and commutativity) formula tree determining $\mathfrak{G}$. We denote this formula tree by $F(\mathfrak{G})$.

Definition 6.1.6. Let $\mathfrak{G} = \langle V, E \rangle$ be a cograph, let $V' \subseteq V$, and let E' be the restriction of E to V'. We say that $\mathfrak{G}' = \langle V', E' \rangle$ is a *subcograph* of $\mathfrak{G}$ iff for all $v \in V'$ and $w_1, w_2 \in V \setminus V'$ we have $vw_1 \in E$ iff $vw_2 \in E$. In this case we also say that V' *induces a subcograph*.

It follows immediately from the definition that any subcograph is indeed a cograph. Furthermore, $\mathfrak{G}'$ is a subcograph of $\mathfrak{G}$ iff $F(\mathfrak{G}')$ is a subformula of $F(\mathfrak{G})$.

Definition 6.1.7. Let $\mathfrak{G} = \langle V, E \rangle$ be a multigraph. A set $B \subseteq E$ of edges is called a *matching* if no two edges in B are adjacent. A matching B is *perfect* if every vertex $v \in V_{\mathfrak{G}}$ is incident to an edge in B. An *R&B-graph* $\mathfrak{G} = \langle V, R, B \rangle$ is a triple such that $\langle V, R \uplus B \rangle$ is a multigraph such that B is a perfect matching and $\langle V, R \rangle$ is a simple graph (i.e., R is not allowed to have multiple edges). We will use the notation $\mathfrak{G}^{\downarrow}$ for the simple graph $\langle V, R \rangle$. An *R&B-cograph* is an R&B-graph $\mathfrak{G} = \langle V, R, B \rangle$ where $\mathfrak{G}^{\downarrow} = \langle V, R \rangle$ is a cograph.

In this presentation we will draw B-edges in blue/bold, and R-edges in red/regular. Below are four examples:


(42)

Definition 6.1.8. A path (resp. cycle) in a graph is said to be *elementary* if it does not contain two equal vertices (resp. but the first and last one). A path $\mathscr{P}$ in a graph with a matching B is *alternating* if the edges of $\mathscr{P}$ are alternately in B and not in B. Let $\mathfrak{G} = \langle V, R, B \rangle$ be an R&B-graph. An *æ-path* in $\mathfrak{G}$ is an elementary alternating path in $\langle V, R \uplus B \rangle$. An *æ-cycle* in $\mathfrak{G}$ is an elementary alternating cycle of even length in $\langle V, R \uplus B \rangle$, so that when turning around the cycle, the edges are still alternately in B and not in B. A *chord* of a path (resp. cycle) is an edge that is not part of the path (resp. cycle) but connects two vertices of the path (resp. cycle). An æ-path (resp. æ-cycle) is called *chordless* iff it does not have any chords.

Note that chords for æ-paths, resp. æ-cycles, are always R-edges because B is a perfect matching. We are now ready to present a central concept for R&B-cographs:

Definition 6.1.9. An R&B-cograph $\mathfrak{G} = \langle V, R, B \rangle$ is *critically chorded* if $\langle V, R \uplus B \rangle$ does not contain any chordless æ-cycle, and any two vertices in V are connected by a chordless æ-path.

In the examples in (42), the first one is not an R&B-cograph, the other three are. The second one has a chordless æ-cycle, and the third one has no chordless æ-path between the lowermost vertices. Only the last one is a critically chorded R&B-cograph.

Definition 6.1.10. Let $\mathfrak{C} = \langle V, R, B \rangle$ be an R&B-graph and $f\colon \mathfrak{C}^{\downarrow} \to \mathfrak{G}$ be a graph-homomorphism and let $\mathfrak{G}$ be $\mathscr{A}$-labeled (where $\mathscr{A}$ is the set of atoms). We say f is *axiom-preserving* iff $xy \in B$ implies that the labels of $f(w)$ and $f(v)$ are dual to each other.

Definition 6.1.11. A graph homomorphism f is a *skew fibration*, denoted as $f\colon \mathfrak{G} \rightarrowtail \mathfrak{G}'$, if for every $v \in V_{\mathfrak{G}}$ and $w' \in V_{\mathfrak{G}'}$ with $f(v)w' \in E'_{\mathfrak{G}}$ there is a $w \in V_{\mathfrak{G}}$ with $vw \in E_{\mathfrak{G}}$ and $f(w)w' \notin E'_{\mathfrak{G}}$.

We are now ready to give the definition of a combinatorial proof together with their main properties.

Definition 6.1.12. A *combinatorial proof* of a sequent Γ consists of a non-empty critically chorded R&B-cograph $\mathfrak{C}$ and an axiom-preserving skew-fibration $f\colon \mathfrak{C}^{\downarrow} \rightarrowtail \mathfrak{G}(\Gamma)$.

Theorem 6.1.13. *A formula is a theorem of classical propositional logic iff it has a combinatorial proof.*

We will later sketch a proof for this theorem. The point to make here is that this theorem is the major distinctive feature between atomic flows and combinatorial proofs. Furthermore, given a map $f\colon \mathfrak{C} \to \mathfrak{G}$ between graphs, it can be checked in time polynomial in the size of the two graphs, whether f is a combinatorial proof. This makes combinatorial proofs an proper proof system in the sense of Cook and Reckhow, and thus play in in the same ligue as the proof systems we have seen in the first section of these lecture notes.

Definition 6.1.14. Let $\mathscr{T}$ be the set of all tautologies. A *proof system* is a surjective **PTIME** function $f\colon X^* \to \mathscr{T}$ where X is some finite alphabet.

In the following sections we study cut elimination for combinatorial proofs and how they are related to deep inference. But first we show how we can draw combinatorial proofs in a way that they can be compared to atomic flows.

Given two sequents Γ and Δ, we write $\phi\colon \Gamma \vdash \Delta$, for a combinatorial proof of the sequent $\bar{\Gamma}, \Delta$. We write $\phi\colon \circ \vdash \Delta$ (resp. $\phi\colon \Gamma \vdash \circ$) if Γ (resp. Δ) is empty.[6] Let ϕ be given by the R&B-cograph $\mathfrak{C}$ and skew fibration $f\colon \mathfrak{C}^{\downarrow} \rightarrowtail \mathfrak{G}(\bar{\Gamma}, \Delta)$. Then the *size* of ϕ, denoted by $\mathsf{size}(\phi)$, is defined to be $\mathsf{size}(\mathfrak{C}^{\downarrow}) + \mathsf{size}(\Gamma) + \mathsf{size}(\Delta)$.

[6] Note that it cannot happen that both Γ and Δ are empty.

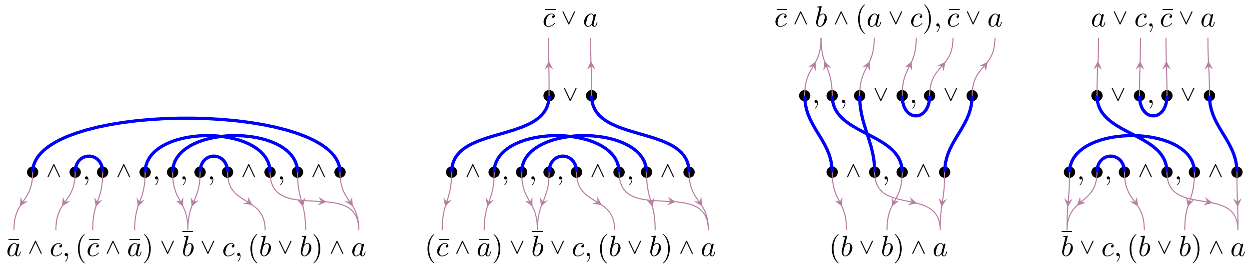


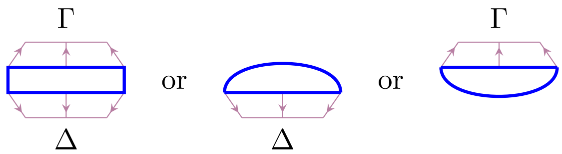
Figure 27: Examples of the same combinatorial proof drawn in different ways

Lemma 6.1.15. *Let $\mathfrak{C}$, $\mathfrak{G}_1$, and $\mathfrak{G}_2$ be cographs and let $f\colon \mathfrak{C} \rightarrowtail \mathfrak{G}_1 \vee \mathfrak{G}_2$ be a skew fibration. Then there are cographs $\mathfrak{C}_1$ and $\mathfrak{C}_2$ and graph homomorphisms $f_1\colon \mathfrak{C}_1 \rightarrowtail \mathfrak{G}_1$ and $f_2\colon \mathfrak{C}_2 \rightarrowtail \mathfrak{G}_2$ such that $\mathfrak{C} = \mathfrak{C}_1 \vee \mathfrak{C}_2$ and $f = f_1 \vee f_2$.*

Proof. This follows immediately from f being a homomorphism. We can let $V_{\mathfrak{C}_1}$ and $V_{\mathfrak{C}_2}$ be the inverse images of $V_{\mathfrak{G}_1}$ and $V_{\mathfrak{G}_2}$, respectively, under f, and let $\mathfrak{C}_1$ and $\mathfrak{C}_2$ be the induced subgraphs. □

Notation 6.1.16. This lemma allows us to depict combinatorial proofs in the following way. Let $\phi\colon \Gamma \vdash \Delta$ be given, let $f\colon \mathfrak{C}^{\downarrow} \rightarrowtail \overline{\mathfrak{G}(\Gamma)} \vee \mathfrak{G}(\Delta)$ be the defining skew fibration, and let $\mathfrak{C}_\Gamma$ and $\mathfrak{C}_\Delta$ be the cographs determined by Lemma 6.1.15 (i.e., $\mathfrak{C}^{\downarrow} = \mathfrak{C}_\Gamma \vee \mathfrak{C}_\Delta$). If we write $F(\overline{\mathfrak{C}_\Gamma})$ and $F(\mathfrak{C}_\Delta)$ for the formula trees corresponding to the cographs $\overline{\mathfrak{C}_\Gamma}$ and $\mathfrak{C}_\Delta$, respectively, then we can write ϕ by writing Γ, $F(\overline{\mathfrak{C}_\Gamma})$, $F(\mathfrak{C}_\Delta)$, and Δ above each other, draw the B-edges and indicate the mapping f by thin (thistle) arrows. Figure 27 shows some examples. For better readability, we allow in $F(\overline{\mathfrak{C}_\Gamma})$ outermost $\wedge$ to be replaced by comma, and in $F(\mathfrak{C}_\Delta)$ outermost $\vee$ to be replaced by comma. Note that the examples in Figure 27 are just "flipped variants" of each other, i.e., are defined by the same R&B-cograph and skew fibration.

Schematically we can depict combinatorial proofs as follows:



where the middle and the right picture are used to indicate that Γ or Δ, respectively, are empty.

Observation 6.1.17. For every formula A, we have a combinatorial proof $1_A\colon A \vdash A$, that we call the *identity* and that is defined by the identity skew fibration $\overline{\mathfrak{G}(A)} \vee \mathfrak{G}(A) \rightarrowtail \mathfrak{G}(\bar{A}, A)$ where the matching is defined such that it pairs each vertex in $V_{\mathfrak{G}(A)}$ to itself in the copy

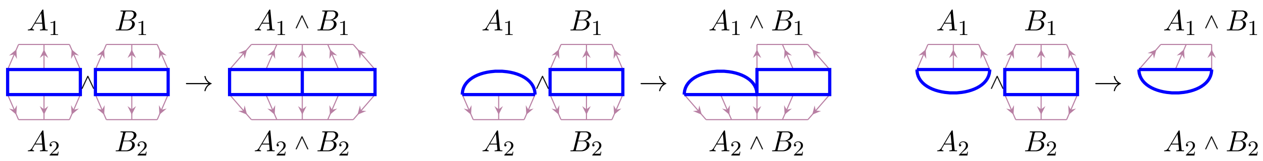
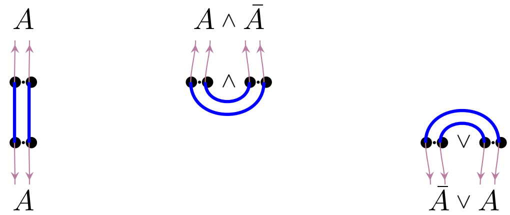


Figure 28: Conjunction of combinatorial proofs

$V_{\overline{\mathfrak{G}(A)}}$. This can we written in the following three ways:



(43)

6.2 Horizontal composition of combinatorial proofs

Lemma 6.2.1. *Let $\phi\colon A_1 \vdash A_2$ and $\psi\colon B_1 \vdash B_2$ be combinatorial proofs. Then there are combinatorial proofs $\chi\colon A_1 \wedge B_1 \vdash A_2 \wedge B_2$ and $\xi\colon A_1 \vee B_1 \vdash A_2 \vee B_2$, such that $\mathsf{size}(\chi) \leq \mathsf{size}(\phi) + \mathsf{size}(\psi)$ and $\mathsf{size}(\xi) \leq \mathsf{size}(\phi) + \mathsf{size}(\psi)$.*

Proof. Let $\mathfrak{C}$ and $\mathfrak{D}$ be the R&B-cographs for ϕ and ψ, respectively, and let $f\colon \mathfrak{C}^{\downarrow} \rightarrowtail \mathfrak{G}(\bar{A}_1) \vee \mathfrak{G}(A_2)$ and $g\colon \mathfrak{D}^{\downarrow} \rightarrowtail \mathfrak{G}(\bar{B}_1) \vee \mathfrak{G}(B_2)$ be their defining skew fibrations. Then, let $\mathfrak{C}_1$ and $\mathfrak{C}_2$ be the subgraphs of $\mathfrak{C}^{\downarrow}$, and $f_1\colon \mathfrak{C}_1 \to \mathfrak{G}(\bar{A}_1)$ and $f_2\colon \mathfrak{C}_2 \to \mathfrak{G}(A_2)$ be the corresponding restrictions of f, obtained via Lemma 6.1.15. Similarly, let $\mathfrak{D}_1$ and $\mathfrak{D}_2$ be the corresponding subgraphs of $\mathfrak{D}^{\downarrow}$, and g_1 and g_2 the corresponding restrictions of g.

The combinatorial proof $\chi\colon A_1 \wedge B_1 \vdash A_2 \wedge B_2$ can now be given by the R&B-cograph $\mathfrak{H}$ and skew fibration $h\colon \mathfrak{H}^{\downarrow} \rightarrowtail \mathfrak{G}(\overline{A_1 \wedge B_1}, A_2 \wedge B_2)$ which are defined as follows:

- If $\mathfrak{C}_2$ and $\mathfrak{D}_2$ are both not empty, then we define $\mathfrak{H}^{\downarrow} = \mathfrak{D}_1 \vee \mathfrak{C}_1 \vee (\mathfrak{C}_2 \wedge \mathfrak{D}_2)$, and $B_{\mathfrak{H}} = B_{\mathfrak{C}} \uplus B_{\mathfrak{D}}$, and $h = g_1 \vee f_1 \vee (f_2 \wedge g_2)$. To see that this is well-defined, note that $\mathfrak{G}(\overline{A_1 \wedge B_1}, A_2 \wedge B_2)$ is the same as $\mathfrak{G}(\bar{B}_1) \vee \mathfrak{G}(\bar{A}_1) \vee (\mathfrak{G}(A_2) \wedge \mathfrak{G}(B_2))$.
- If $\mathfrak{C}_2$ is empty then $\mathfrak{C}_1 = \mathfrak{C}^{\downarrow}$ and we define $\mathfrak{H} = \mathfrak{C}$ and let h behave as f does.
- If $\mathfrak{D}_2$ is empty and $\mathfrak{C}_2$ is not, then then $\mathfrak{D}_1 = \mathfrak{D}^{\downarrow}$ and we define $\mathfrak{H} = \mathfrak{D}$ and let h behave as g does.

Then, $\mathfrak{H}$ is an R&B-cograph (by construction) and it is critically chorded. It also trivially follows that h is axiom preserving. Therefore it only remains to show that h is indeed a skew fibration. For this, observe that $g_1 \vee f_1 \vee (f_2 \wedge g_2)$ fails to be a skew fibration only if one of $\mathfrak{C}_2$ or $\mathfrak{D}_2$ is empty. On the other hand, f is a skew-fibration from $\mathfrak{C}^{\downarrow}$ to $\mathfrak{G}(\bar{B}_1) \vee \mathfrak{G}(\bar{A}_1) \vee (\mathfrak{G}(A_2) \wedge \mathfrak{G}(B_2))$ if no vertex of $\mathfrak{C}$ is mapped to $\mathfrak{G}(A_2)$, i.e., $\mathfrak{C}_2$ is empty.

Dually, we can define $\xi\colon A_1 \vee B_1 \vdash A_2 \vee B_2$. $\square$

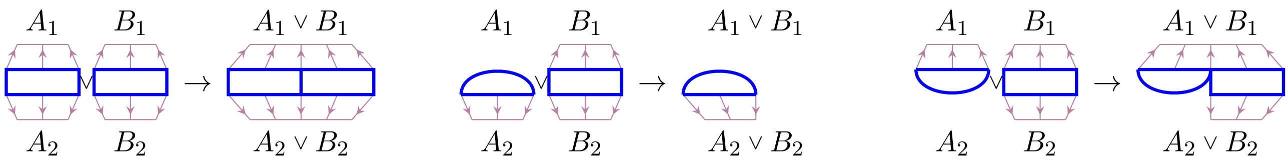


Figure 29: Disjunction of combinatorial proofs

Remark 6.2.2. Note that it is crucial to check whether $\mathfrak{C}_2$ or $\mathfrak{D}_2$ are empty, whereas for $\mathfrak{C}_1$ and $\mathfrak{D}_1$, this is irrelevant. The difference is shown in Figure 28 (and dually in Figure 29). Note also that there is an arbitrary choice to make when both $\mathfrak{C}_2$ and $\mathfrak{D}_2$ are empty.

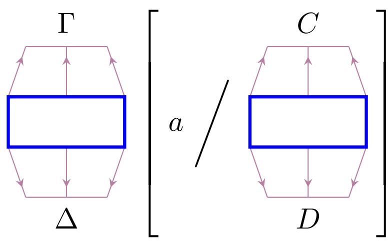
6.3 Substitution for combinatorial proofs

Usually substitution means to replace in a formula a variable by another formula. With combinatorial proofs, we can bring this to the next level: replacing inside a proof a variable by another proof.

Definition 6.3.1. A *substitution* is a mapping σ from propositional variables to formulas such that $\sigma(a) \neq a$ for only finitely many a.

We write $A\sigma$ for the formula obtained from applying the substitution σ to the formula A. If $\sigma = \{a_1 \mapsto B_1, \ldots, a_n \mapsto B_n\}$ we also write $A[a_1/B_1, \ldots, a_n/B_n]$ for $A\sigma$. This normally means that not only is each occurrence of a_i in A is replaced by B_i in A, but also each occurrence of $\bar{a}_i$ is replaced by $\bar{B}_i$. We also need a notation for substitutions in which an variable a and its dual $\bar{a}$ are not replaced by dual formulas. In this case we write $A[a_1/B_1, \bar{a}_1/C_1, \ldots, a_n/B_n, \bar{a}_n/C_n]$ for the formula that is obtained from A by simultaneously replacing every a_i by B_i and every $\bar{a}_i$ by C_i for each $i \in \{1, \ldots, n\}$.

Now we define the substitution $\phi[a/\psi] \colon \Gamma\big[a/C, \bar{a}/\bar{D}\big] \vdash \Delta\big[a/D, \bar{a}/\bar{C}\big]$ shown below



(44)

The basic idea of the construction is as follows: The combinatorial proof $\phi \colon \Gamma \vdash \Delta$ consists of simple paths, and each simple path in ϕ whose endpoints are occurrences of a or $\bar{a}$ are replaced according to Figure 30. To define this more formally, we first need the notion of substitution in a graph.

Construction 6.3.2. Let $\mathfrak{C}$ and $\mathfrak{D}$ be disjoint graphs, and let x be a vertex in $\mathfrak{C}$. With $\mathfrak{C}[x/\mathfrak{D}]$ we denote the graph whose vertex set is $V = V_{\mathfrak{C}} \setminus \{x\} \cup V_{\mathfrak{D}}$ and whose edge set is $E = E_{\mathfrak{C}} \setminus \{xz \mid z \in V_{\mathfrak{C}}\} \cup \{yz \mid y \in V_{\mathfrak{D}}, xz \in E_{\mathfrak{C}}\}$. In other words, we remove x from $\mathfrak{C}$ and replace it by $\mathfrak{D}$, such that we have an edge from a remaining vertex y in $\mathfrak{C}$ to all vertices in $\mathfrak{D}$, whenever there was an edge from y to x in $\mathfrak{C}$ before.

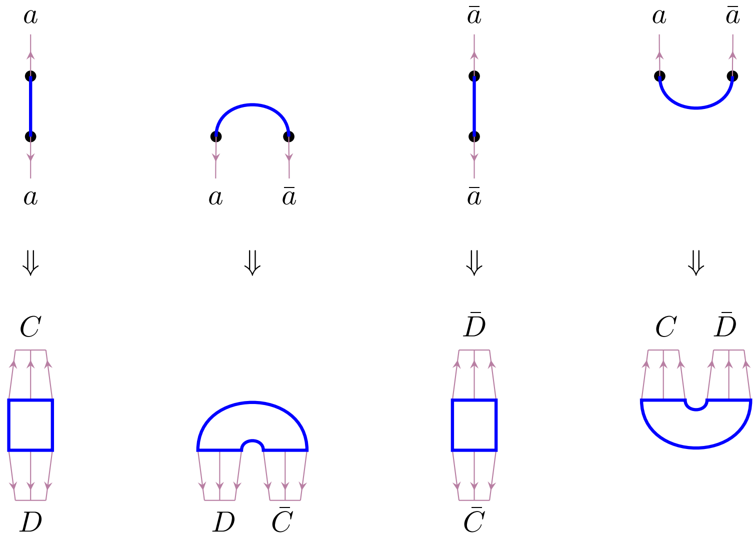


Figure 30: Substitution of combinatorial proofs

Lemma 6.3.3. *If $\mathfrak{C}$ and $\mathfrak{D}$ are cographs and $x \in V_{\mathfrak{C}}$, then $\mathfrak{C}[x/\mathfrak{D}]$ is also a cograph.*

Proof. If we take the formula tree for $\mathfrak{C}$, remove the leaf x, and replace it by the formula tree of $\mathfrak{D}$, we obtain a formula tree for $\mathfrak{C}[x/\mathfrak{D}]$, which is therefore a cograph by Proposition 6.1.5. □

Construction 6.3.4. In Construction 6.3.2 we substituted graphs for *vertexes* in other graphs. Now we use this to substitute R&B-graphs for *B-edges* in other R&B-graphs. Let $\mathfrak{C}$ and $\mathfrak{D}$ be disjoint R&B-graphs, and let $x, y \in V_{\mathfrak{C}}$ with $xy \in B_{\mathfrak{C}}$. Furthermore, let $\mathfrak{D}^{\downarrow} = \mathfrak{D}_1 \vee \mathfrak{D}_2$. We now define the R&B-graph $\mathfrak{H} = \mathfrak{C}[xy/\langle \mathfrak{D}_1 \vee \mathfrak{D}_2, B_{\mathfrak{D}}\rangle] = \langle V_{\mathfrak{H}}, R_{\mathfrak{H}}, B_{\mathfrak{H}}\rangle$ as follows. We let $\langle V_{\mathfrak{H}}, R_{\mathfrak{H}}\rangle = \mathfrak{C}^{\downarrow}[x/\mathfrak{D}_1][y/\mathfrak{D}_2]$, applying Construction 6.3.2 twice, and let $B_{\mathfrak{H}} = B_{\mathfrak{C}} \setminus \{xy\} \cup B_{\mathfrak{D}}$. In other words, x is replaced by $\mathfrak{D}_1$ and y by $\mathfrak{D}_2$, and the B-edge xy is removed an replaced by the matching $B_{\mathfrak{D}}$.

Lemma 6.3.5. *If $\mathfrak{C}$ and $\mathfrak{D}$ are R&B-cographs with $xy \in B_{\mathfrak{C}}$ and $\mathfrak{D}^{\downarrow} = \mathfrak{D}_1 \vee \mathfrak{D}_2$ then $\mathfrak{H} = \mathfrak{C}[xy/\langle \mathfrak{D}_1 \vee \mathfrak{D}_2, B_{\mathfrak{D}}\rangle]$ also is an R&B-cograph. Furthermore, if $\mathfrak{C}$ and $\mathfrak{D}$ are both critically chorded, then so is $\mathfrak{H}$.*

Proof. The graph $\mathfrak{H}$ is a cograph for the same reason as in Lemma 6.3.3. Now assume by way of contradiction that $\mathfrak{H}$ is not critically chorded. First, assume there is a chordless æ-cycle $\mathscr{C}$. If all vertices of $\mathscr{C}$ are inside $V_{\mathfrak{C}}$ or all inside $V_{\mathfrak{D}}$, we have immediately a contradiction to $\mathfrak{C}$ and $\mathfrak{D}$ having no chordless æ-cycle. So, the cycle $\mathscr{C}$ must contain vertices from $V_{\mathfrak{C}}$ and $V_{\mathfrak{D}}$. Since by construction all B-edges are fully contained in $\mathfrak{C}$ or in $\mathfrak{D}$, we must have an R-edge participating in $\mathscr{C}$ and connecting a vertex $u \in V_{\mathfrak{C}}$ to a vertex $z \in V_{\mathfrak{D}}$. Let $v \in V_{\mathfrak{C}}$ be the unique vertex with $uv \in B_{\mathfrak{C}}$. However, since $uz \in R_{\mathfrak{H}}$, we must by construction also have $vz \in R_{\mathfrak{H}}$ which is a chord for $\mathscr{C}$. Contradiction. For showing that that any two vertices in $\mathfrak{H}$ are connected by a chordless path, we can proceed similarly. □

Lemma 6.3.6. *Let $\phi\colon \Gamma \vdash \Delta$ and $\psi\colon C \vdash D$ be combinatorial proofs. Then there is a combinatorial proof $\phi'\colon \Gamma[a/C, \bar{a}/\bar{D}] \vdash \Delta[a/D, \bar{a}/\bar{C}]$.*

Proof. Let ϕ and ψ as above and let $\Gamma' = \Gamma\big[a/C, \bar{a}/\bar{D}\big]$ and $\Delta' = \Delta\big[a/D, \bar{a}/\bar{C}\big]$. For constructing $\phi' \colon \Gamma' \vdash \Delta'$, let $\mathfrak{C}$ and $\mathfrak{D}$ be the R&B-cographs for ϕ and ψ, respectively, and let $f \colon \mathfrak{C}^{\downarrow} \rightarrowtail \mathfrak{G}(\bar{\Gamma}, \Delta)$ and $g \colon \mathfrak{D}^{\downarrow} \rightarrowtail \mathfrak{G}(\bar{C}, D)$ be their corresponding skew fibrations. For brevity, we write $\mathfrak{G}$ for $\mathfrak{G}(\bar{\Gamma}, \Delta)$, and $\mathfrak{G}'$ for $\mathfrak{G}(\bar{\Gamma}', \Delta')$. Next, let $\mathfrak{D}_{\bar{C}}$ and $\mathfrak{D}_D$ be the two cographs obtained from $\mathfrak{D}^{\downarrow}$ via Lemma 6.1.15, and let $x_1, \ldots, x_n \in V_{\mathfrak{C}}$ be the vertexes that f maps to a vertex labeled $\bar{a}$ in $\mathfrak{G}$, and and let $y_1, \ldots, y_n \in V_{\mathfrak{C}}$ be all the vertexes that f maps to a vertex labeled a in $\mathfrak{G}$ — their number has to be identical, otherwise f could not be axiom preserving. Without loss of generality, we can assume that $\{x_1y_1, \ldots, x_ny_n\} \subseteq B_{\mathfrak{C}}$. We can now give the R&B-cograph $\mathfrak{C}'$ for ϕ' as follows:

$$\mathfrak{C}' = \mathfrak{C}[x_1y_1/\langle \mathfrak{D}_{\bar{C}} \vee \mathfrak{D}_D, B_{\mathfrak{D}} \rangle] \cdots [x_ny_n/\langle \mathfrak{D}_{\bar{C}} \vee \mathfrak{D}_D, B_{\mathfrak{D}} \rangle]$$

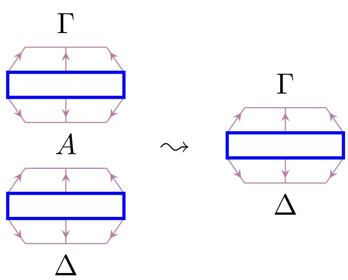
applying Construction 6.3.4 for each B-edge in $\mathfrak{C}$ connecting an a and an $\bar{a}$ in $\mathfrak{G}$. Finally, we define the map $f' \colon \mathfrak{C}' \rightarrowtail \mathfrak{G}'$ as follows: For every $z \in V_{\mathfrak{C}} \setminus \{x_1, \ldots, x_n, y_1, \ldots, y_n\}$, we have $f'(z) = f(z)$. For each x_i that is mapped by f to a $\bar{a}$, we use g to map the substituted copy of $\mathfrak{D}_{\bar{C}}$ in $\mathfrak{C}'$ to the corresponding substituted copy of $\mathfrak{G}(\bar{C})$ in $\mathfrak{G}'$. We proceed similarly for each y_i. It is easy to see that the so defined f' is indeed a skew fibration and axiom preserving. $\square$

6.4 Vertical composition of combinatorial proofs

In this section, we finally show what cuts in combinatorial proofs are, and how they can be eliminated.

Theorem 6.4.1. *Let $\phi \colon \Gamma \vdash A$ and $\psi \colon A \vdash \Delta$ be combinatorial proofs. Then there is a combinatorial proof $\chi \colon \Gamma \vdash \Delta$.*

Before we can give the construction of χ, as indicated below:



(45)

we need first to establish some preliminary properties on skew fibrations and the composition of R&B-cographs.

Lemma 6.4.2. *Let $\mathfrak{C}$, $\mathfrak{D}$, $\mathfrak{G}$, $\mathfrak{H}$ be cographs.*

1. *If $f \colon \mathfrak{C} \rightarrowtail \mathfrak{G}$ is an isomorphism, then it is also a skew fibration.*

2. *The map $\mathsf{w} \colon \mathfrak{C} \rightarrowtail \mathfrak{C} \vee \mathfrak{D}$, which behaves like the identity on $\mathfrak{C}$, is a skew fibration.*

3. *The map $\mathsf{c} \colon \mathfrak{C} \vee \mathfrak{C} \rightarrowtail \mathfrak{C}$, which maps both copies of $\mathfrak{C}$ in the domain like the identity to the $\mathfrak{C}$ in the codomain, is a skew fibration.*

4. *The map* $\mathsf{m}\colon (\mathfrak{C} \wedge \mathfrak{D}) \vee (\mathfrak{G} \wedge \mathfrak{H}) \rightarrowtail (\mathfrak{C} \vee \mathfrak{G}) \wedge (\mathfrak{D} \vee \mathfrak{H})$*, which maps each* $\mathfrak{C}$*,* $\mathfrak{D}$*,* $\mathfrak{G}$*,* $\mathfrak{H}$ *identically to itself, is a skew fibration.*

5. *If* $f\colon \mathfrak{C} \rightarrowtail \mathfrak{G}$ *and* $g\colon \mathfrak{D} \rightarrowtail \mathfrak{H}$ *are skew fibrations, then so are* $f \vee g\colon \mathfrak{C} \vee \mathfrak{D} \rightarrowtail \mathfrak{G} \vee \mathfrak{H}$ *and* $f \wedge g\colon \mathfrak{C} \wedge \mathfrak{D} \rightarrowtail \mathfrak{G} \wedge \mathfrak{H}$*.*

6. *If* $f\colon \mathfrak{C} \rightarrowtail \mathfrak{G}$ *and* $g\colon \mathfrak{G} \rightarrowtail \mathfrak{H}$ *are skew fibrations, then so is* $g \circ f\colon \mathfrak{C} \rightarrowtail \mathfrak{H}$*.*

Exercise 6.4.3. Prove this lemma.

Construction 6.4.4. Let $\mathfrak{C}$ and $\mathfrak{D}$ be R&B-cographs such that $\mathfrak{C}^{\downarrow} = \mathfrak{G} \vee \mathfrak{H}$ and $\mathfrak{D}^{\downarrow} = \bar{\mathfrak{H}} \vee \mathfrak{K}$ for some cographs $\mathfrak{G}$, $\mathfrak{H}$, and $\mathfrak{K}$. We define the graph $\mathfrak{B} = \langle V_{\mathfrak{B}}, E_{\mathfrak{B}} \rangle$ with $V_{\mathfrak{B}} = V_{\mathfrak{G}} \uplus V_{\mathfrak{H}} \uplus V_{\mathfrak{K}}$ and $E_{\mathfrak{B}} = B_{\mathfrak{C}} \uplus B_{\mathfrak{D}}$. This allows us to define the R&B-cograph $\mathfrak{E} = \mathfrak{C} \diamond \mathfrak{D}$ as follows: We let $\mathfrak{E}^{\downarrow} = \mathfrak{G} \vee \mathfrak{K}$, i.e., $V_{\mathfrak{E}} = V_{\mathfrak{G}} \cup V_{\mathfrak{K}}$ and $R_{\mathfrak{E}} = E_{\mathfrak{G}} \cup E_{\mathfrak{K}}$, and we let $xy \in B_{\mathfrak{E}}$ iff there is a path from x to y in $\mathfrak{B}$. Note that this indeed defines a perfect matching. For each x in $V_{\mathfrak{E}}$ there is a unique y connected to x by a path in $\mathfrak{B}$ because $B_{\mathfrak{C}}$ and $B_{\mathfrak{D}}$ are both perfect matchings.

Lemma 6.4.5. *If in Construction 6.4.4 the R&B-cographs* $\mathfrak{C}$ *and* $\mathfrak{D}$ *are critically chorded, then so is* $\mathfrak{E} = \mathfrak{C} \diamond \mathfrak{D}$*.*

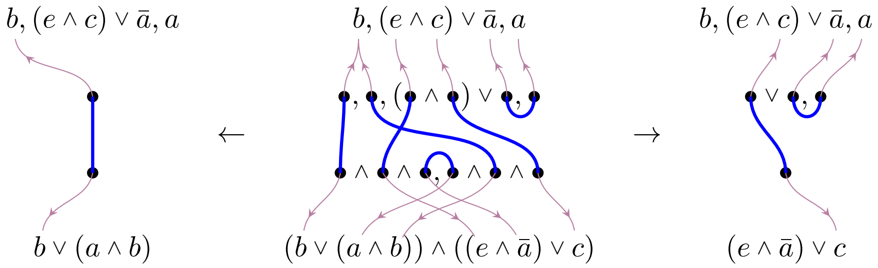
Next, we define for a combinatorial proof $\phi\colon \Gamma \vdash B \wedge C$ the two *projections* $\phi_l\colon \Gamma \vdash B$ and $\phi_r\colon \Gamma \vdash C$ that "forget" the information about the deleted subformula. Their existence should not be surprising since from a proof of $B \wedge C$ one should be able to recover proofs of B and of C from the same premises.

Construction 6.4.6. Let $\phi\colon \Gamma \vdash B \wedge C$ be given by the R&B-cograph $\mathfrak{C}$ and the skew fibration $f\colon \mathfrak{C}^{\downarrow} \rightarrowtail \mathfrak{G}(\bigwedge\bar{\Gamma}) \vee (\mathfrak{G}(B) \wedge \mathfrak{G}(C))$. Let $U_C \subseteq V_{\mathfrak{C}}$ be the set of all vertices in $\mathfrak{C}$ that are mapped by f to atom occurrences in C, and let $U_C^{\perp} \subseteq V_{\mathfrak{C}}$ be the smallest set such that

- If $x \in U_C$ and $xy \in B_{\mathfrak{C}}$ and $y \notin U_C$ then $y \in U_C^{\perp}$.
- If $x \in U_C^{\perp}$ and $xy \in B_{\mathfrak{C}}$ and $y \notin U_C$ then $y \in U_C^{\perp}$.
- If $V', V'' \subseteq V_{\mathfrak{C}}$ induce subcographs and $V' \subseteq U_C^{\perp}$ and $V' \cap V'' = \emptyset$ and $V' \cup V''$ induces a subcograph such that for all $v' \in V'$ and $v'' \in V''$ we have $v'v'' \in R_{\mathfrak{C}}$, then also $V'' \subseteq U_C^{\perp}$.

Now let $V_{\mathfrak{C}_l} = V \setminus (U_C \cup U_C^{\perp})$, and let $R_{\mathfrak{C}_l}$ and $B_{\mathfrak{C}_l}$ be the restrictions of $R_{\mathfrak{C}}$ and $B_{\mathfrak{C}}$ (respectively) to $V_{\mathfrak{C}_l}$. Finally, we can define $\phi_l\colon \Gamma \vdash B$ by $\mathfrak{C}_l = \langle V_{\mathfrak{C}_l}, R_{\mathfrak{C}_l}, B_{\mathfrak{C}_l} \rangle$ and $f_l\colon \mathfrak{C}_l^{\downarrow} \rightarrowtail \mathfrak{G}(\bigwedge\bar{\Gamma}) \vee \mathfrak{G}(B)$ which is f restricted to $V_{\mathfrak{C}_l}$.

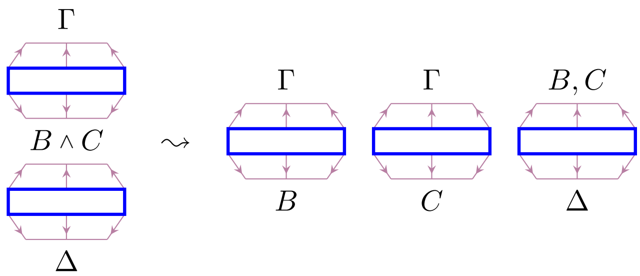
It is easy to see that $\mathfrak{C}_l$ is critically chorded: any chordless æ-cycle would already be present in $\mathfrak{C}$, and any two vertices are connected by the same chordless æ-path as in $\mathfrak{C}$. We also have that $V_{\mathfrak{C}_l} \neq \emptyset$ (otherwise there would be a chordless æ-cycle in $\mathfrak{C}$.). Finally, it is easy to see that f_l is axiom preserving and a skew fibration. Thus, $\phi_l\colon \Gamma \vdash B$ is indeed a combinatorial proof. In the same way we can define the right projection $\phi_r \colon \Gamma \vdash C$. Below is an example of a combinatorial proof and its two projections:



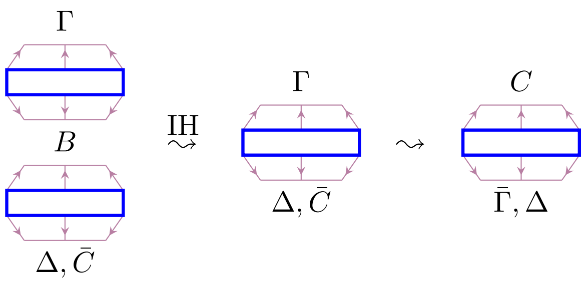
(46)

In a dual way, we can define for a combinatorial proof $\psi \colon B \vee C \vdash \Delta$ its left and right projections $\psi_l \colon B \vdash \Delta$ and $\psi_r \colon C \vdash \Delta$.

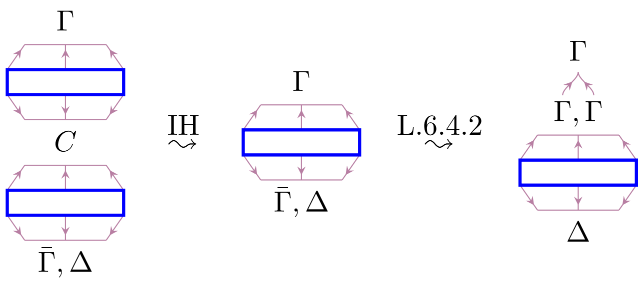
Proof of Theorem 6.4.1. We proceed by induction on the formula A. First, assume $A = B \wedge C$. Then, from $\phi \colon \Gamma \vdash B \wedge C$ we can obtain the two projections $\phi_l \colon \Gamma \vdash B$ and $\phi_r \colon \Gamma \vdash C$, and from $\psi \colon B \wedge C \vdash \Delta$, we get $\psi' \colon B, C \vdash \Delta$:



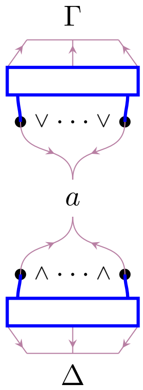
From ψ' we can obtain $\psi'' \colon B \vdash \Delta, \bar{C}$, which can be composed with ϕ_l to get, by induction hypothesis, $\xi \colon \Gamma \vdash \Delta, \bar{C}$, from which we can get $\chi' \colon C \vdash \bar{\Gamma}, \Delta$:



This can be composed with ϕ_r, which gives us by induction hypothesis a combinatorial proof $\chi'' \colon \Gamma \vdash \bar{\Gamma}, \Delta$, from which we get a $\chi' \colon \Gamma, \Gamma \vdash \Delta$. Finally, we can apply Lemma 6.4.2 to get the desired $\chi \colon \Gamma \vdash \Delta$:



If $A = B \vee C$ we proceed analogous. It remains to show the case when A is an atom, i.e., we have the situation:

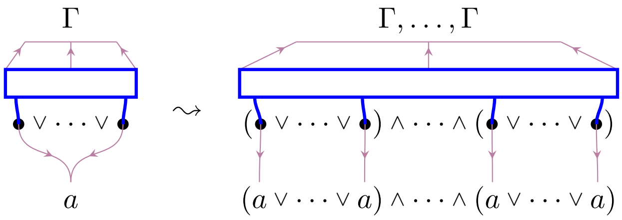


(47)

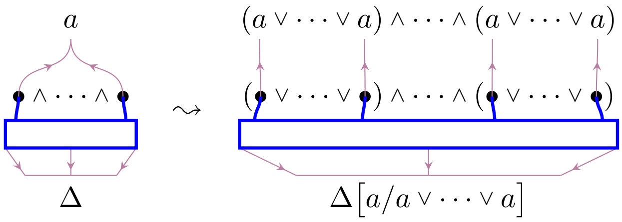
Let $f\colon \mathfrak{C}^{\downarrow} \rightarrowtail \mathfrak{G}(\bigwedge\bar{\Gamma}, a)$ and $g\colon \mathfrak{D}^{\downarrow} \rightarrowtail \mathfrak{G}(\bar{a}, \bigvee\Delta)$ be the skew fibrations of the combinatorial proofs $\phi\colon \Gamma \vdash a$ and $\psi\colon a \vdash \Delta$, respectively. Let $x_1, \ldots, x_n$ be the vertices in $\mathfrak{C}$ that are mapped by f to the a in the conclusion of ϕ, and let $y_1, \ldots, y_m$ be the vertices in $\mathfrak{D}$ that are mapped by g to the occurrence of $\bar{a}$ that represents the a in the premise of ψ.

Now we define the map $f^*\colon \mathfrak{C}^{\downarrow} \rightarrowtail \mathfrak{G}(\bigwedge\bar{\Gamma}, a \vee \cdots \vee a)$ where we replace a by a disjunction of n copies of a, and let f^* behave as f on $V_{\mathfrak{C}} \setminus \{x_1, \ldots, x_n\}$ and map each x_i to one copy of a. This clearly also is a skew fibration, and in a similar way we define the skew fibration $g^*\colon \mathfrak{D}^{\downarrow} \rightarrowtail \mathfrak{G}(\bar{a} \vee \cdots \vee \bar{a}, \bigvee\Delta)$ where we use m copies of $\bar{a}$. We let $\phi^*\colon \Gamma \vdash a \vee \cdots \vee a$ and $\psi^*\colon a \wedge \cdots \wedge a \vdash \Delta$ be the combinatorial proofs defined by f^* and g^*, respectively.

We now apply the construction of Section 6.2 to form the conjunction of m copies of ϕ^*, which yields a combinatorial proof $\hat{\phi}\colon \Gamma, \ldots, \Gamma \vdash (a \vee \cdots \vee a) \wedge \cdots \wedge (a \vee \cdots \vee a)$, as indicated below:



Next, we substitute in ψ^* all paths that start in the premise $a \wedge \cdots \wedge a$ by the identity $1\colon a \vee \cdots \vee a \vdash a \vee \cdots \vee a$ (with m copies of a on each side) as done in Section 6.3. Then we have a combinatorial proof $\hat{\psi}\colon (a \vee \cdots \vee a) \wedge \cdots \wedge (a \vee \cdots \vee a) \vdash \Delta[a/a \vee \cdots \vee a]$ as shown below:



We now plug $\hat{\phi}$ and $\hat{\psi}$ together and apply Lemma 6.4.5 to get $\chi'\colon \Gamma, \ldots, \Gamma \vdash \Delta[a/a \vee \cdots \vee a]$,

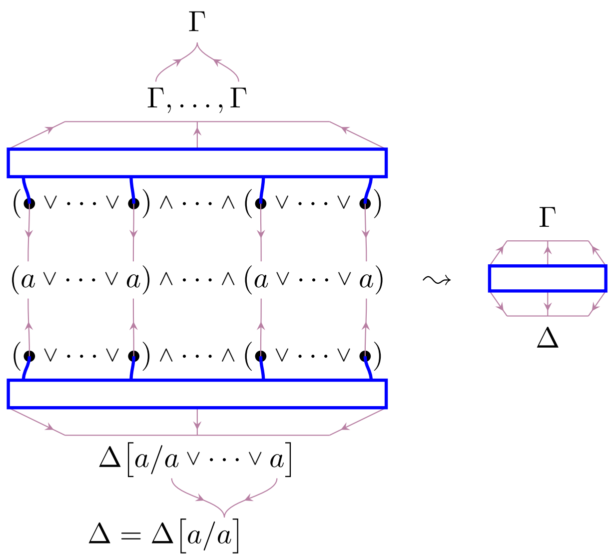
$$
\mathsf{ai}\!\downarrow \frac{A}{A \wedge (a \vee \bar{a})} \qquad \mathsf{s}\,\frac{(A \vee B) \wedge C}{A \vee (B \wedge C)} \qquad \mathsf{ai}\!\uparrow \frac{(\bar{a} \wedge a) \vee A}{A}
$$

$$
\mathsf{ac}\!\downarrow \frac{a \vee a}{a} \qquad \mathsf{m}\,\frac{(A \wedge C) \vee (B \wedge D)}{(A \vee B) \wedge (C \vee D)} \qquad \mathsf{ac}\!\uparrow \frac{a}{a \wedge a}
$$

$$
\mathsf{w}\!\downarrow \frac{A}{A \vee B} \qquad \mathsf{w}\!\uparrow \frac{B \wedge A}{A}
$$

Figure 31: Deep inference system SKS

to which we apply Lemma 6.4.2



to get the desired combinatorial proof $\chi \colon \Gamma \vdash \Delta$. □

6.5 Relation to to deep inference proofs

Let us now show hoe combinatorial proofs are related to syntactic proofs given in a deep inference formalism. For simplicity, we use here a variant of SKS without the units t and f, as shown in Figure 31, and we work modulo the equivalence relation defined by associativity and commutativity of $\wedge$ and $\vee$, as given in (40).

Each rule in system SKS can straightforwardly be translated into a combinatorial proof, as indicated in Figure 32, where the double lines indicate the identity (see Observation 6.1.17). Note that for the m-rule there are two possible translations. Since whenever $A = B$ modulo associativity and commutativity (40) we have that $\mathfrak{G}(A) = \mathfrak{G}(B)$, an equivalence step in an SKS-proof can translated into the identity proof. This is enough to give a direct translation from a SKS derivation $\Phi \colon A \to B$, to a sequence of combinatorial proofs composed by cut, to which we have to apply Theorem 6.4.1 to get a combinatorial proof $\phi \colon A \vdash B$.

Here is the main theorem on the relation between SKS and combinatorial proofs:

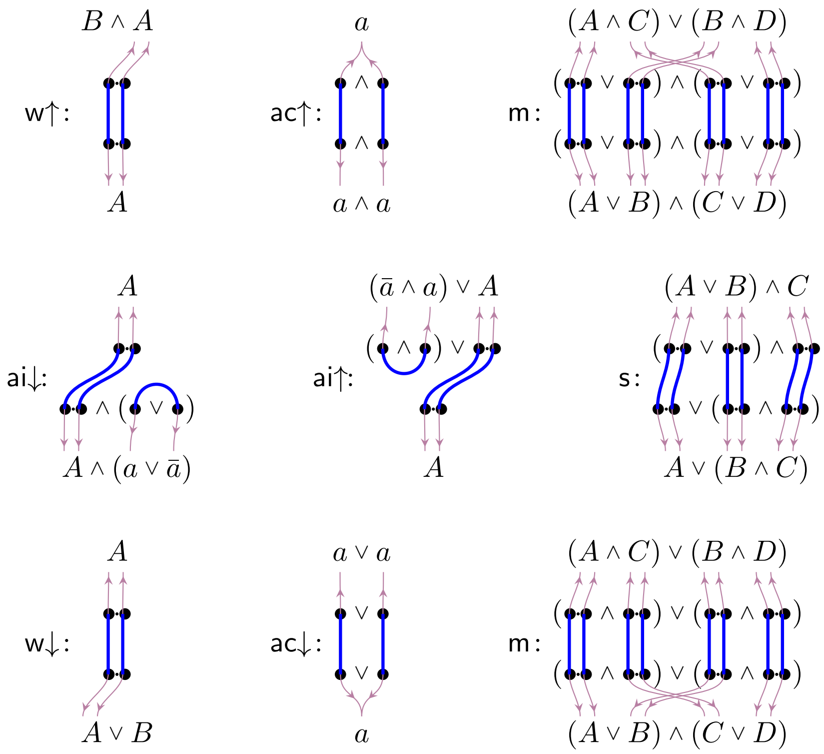
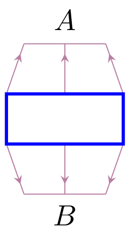
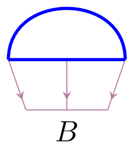


Figure 32: Simple combinatorial flows for the rules in Figure 31



$\Leftrightarrow$

$$\begin{array}{c} A \\ {\scriptstyle \{\mathsf{w}\uparrow,\mathsf{ac}\uparrow,\mathsf{m}\}}\,\Big\Vert\,\Phi_1 \\ A' \\ {\scriptstyle \{\mathsf{ai}\downarrow,\mathsf{ai}\uparrow,\mathsf{s}\}}\,\Big\Vert\,\Phi_2 \\ B' \\ {\scriptstyle \{\mathsf{w}\downarrow,\mathsf{ac}\downarrow,\mathsf{m}\}}\,\Big\Vert\,\Phi_3 \\ B \end{array}$$



$\Leftrightarrow$

$$\begin{array}{c} {\scriptstyle \{\mathsf{ai}\downarrow,\mathsf{s}\}}\,\Big\Vert\,\Psi_2 \\ B' \\ {\scriptstyle \{\mathsf{w}\downarrow,\mathsf{ac}\downarrow,\mathsf{m}\}}\,\Big\Vert\,\Psi_3 \\ B \end{array}$$

Figure 33: Statements of Theorem 6.5.1

Theorem 6.5.1. *Let A and B be formulas. There is a combinatorial proof $\phi\colon A \to B$ iff there are formulas A' and B' such that there are derivations*

$$\begin{array}{c} A \\ {\scriptstyle \{\mathsf{w}\uparrow,\mathsf{ac}\uparrow,\mathsf{m}\}}\,\Big\Vert\,\Phi_1 \\ A' \end{array} \quad \textit{and} \quad \begin{array}{c} A' \\ {\scriptstyle \{\mathsf{ai}\downarrow,\mathsf{ai}\uparrow,\mathsf{s}\}}\,\Big\Vert\,\Phi_2 \\ B' \end{array} \quad \textit{and} \quad \begin{array}{c} B' \\ {\scriptstyle \{\mathsf{w}\downarrow,\mathsf{ac}\downarrow,\mathsf{m}\}}\,\Big\Vert\,\Phi_3 \\ B \end{array}$$

and such that $\mathsf{size}(\phi) = O(\mathsf{size}(\Phi_1) + \mathsf{size}(\Phi_2) + \mathsf{size}(\Phi_3))$*. Similarly, there is a combinatorial proof $\psi\colon \circ \vdash B$ iff there are derivations*

$$\begin{array}{c} {\scriptstyle \{\mathsf{ai}\downarrow,\mathsf{s}\}}\,\Big\Vert\,\Psi_2 \\ B' \end{array} \quad \textit{and} \quad \begin{array}{c} B' \\ {\scriptstyle \{\mathsf{w}\downarrow,\mathsf{ac}\downarrow,\mathsf{m}\}}\,\Big\Vert\,\Psi_3 \\ B \end{array}$$

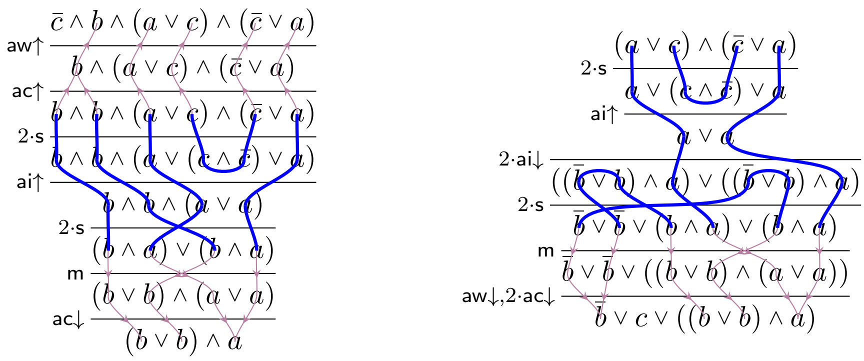


Figure 34: Examples for Theorem 6.5.1

such that $\mathsf{size}(\psi) = O(\mathsf{size}(\Psi_2) + \mathsf{size}(\Psi_3))$

Both statements are indicated in Figure 33.

Exercise 6.5.2. Show how Theorem 6.1.13 can be proved using Theorem 6.5.1.

The proof of Theorem 6.5.1 essentially consists of the following two lemmas:

Lemma 6.5.3. *Let A and B be formulas. There is a skew fibration $f\colon \mathfrak{G}(A) \rightarrowtail \mathfrak{G}(B)$ iff there is a derivation Φ from A to B in* $\{\mathsf{w}\!\downarrow, \mathsf{ac}\!\downarrow, \mathsf{m}\}$.

Lemma 6.5.4. *Let A and B be formulas. There is a critically chorded R&B-graph $\mathfrak{C}$ with $\mathfrak{C}^{\downarrow} = \mathfrak{G}(\bar{A}) \vee \mathfrak{G}(B)$ iff there is a derivation Φ from A to B in* $\{\mathsf{ai}\!\downarrow, \mathsf{ai}\!\uparrow, \mathsf{s}\}$.

Proving these two lemmas would take us too far out of the scope of these course notes. But we give some ideas in the notes below.

Proof of Theorem 6.5.1. First, assume we have an SKS derivation as shown on the left in Figure 33. We let $\mathfrak{C}^{\downarrow} = \mathfrak{G}(\bar{A}') \vee \mathfrak{G}(B')$. By Lemma 6.5.4 we have a matching $B_{\mathfrak{C}}$ such that the R&B-graph $\mathfrak{C}$ is critically chorded. Then we can form the SKS derivation using only rules $\mathsf{w}\!\downarrow$, $\mathsf{ac}\!\downarrow$ and m from $\bar{A}' \vee B'$ to $\bar{A} \vee B$ by horizontally composing the dual of Φ_1 with Φ_3. By Lemma 6.5.3 we get our skew fibration $f\colon \mathfrak{C}^{\downarrow} \rightarrowtail \mathfrak{G}(\bar{A}) \vee \mathfrak{G}(B)$. Conversely, let ϕ be given, let $f\colon \mathfrak{C}^{\downarrow} \rightarrowtail \mathfrak{G}(\bar{A}, B)$ be its skew fibration, and let $\mathfrak{C}_{\bar{A}}$ and $\mathfrak{C}_B$ be the two cographs obtained via Lemma 6.1.15. If we add labels to $\mathfrak{C}_{\bar{A}}$ and $\mathfrak{C}_B$ such that f is label-preserving, we can let A' and B' be the formulas determined by $\overline{\mathfrak{C}_{\bar{A}}}$ and $\mathfrak{C}_B$, respectively. We can now apply Lemma 6.5.4 to get Φ_2, and Lemma 6.5.3 to get Φ_3 and (the dual of) Φ_1. $\square$

Figure 34 shows two examples of derivations enriched with the "flow-graph" tracing the atoms in the derivation. The corresponding combinatorial proofs are in Figure 27.

Exercise 6.5.5. Which combinatorial proofs in Figure 27 correspond to the derivations in Figure 34? Find similar derivations for the others.

Exercise 6.5.6. Prove the second decomposition thereom (Theorem 4.1.8) using Theorem 6.5.1 (and Theorem 6.4.1).

6.6 Notes

The original work on combinatorial proofs by Dominic Hughes is [Hug06a] and [Hug06b]. The notion of R&B-cograph goes back to Retoré's PhD-thesis [Ret93, Ret03]. The relation between cographs and formulas is even older and can already been found in [Duf65], see also [Möh89].

Our presentation of the condition on the cograph in a combinatorial proof deters from Hughes' [Hug06a] and follows Retoré's [Ret03] instead. The reason is that Retoré makes the relation to proof nets of linear logic [DR89] explicit. Furthermore, the condition on the cograph $\mathfrak{C}^{\downarrow}$ given by Hughes [Hug06a, Hug06b] is weaker than ours. It is equivalent to our condition of $\mathfrak{C}$ not containing any chordless æ-cycle. In terms of linear logic, this is equivalent to the correctness condition for MLL proof nets with the mix-rule [Ret03]:

$$\mathsf{mix}\,\frac{\Gamma \quad \Delta}{\Gamma, \Delta}$$

In our presentation here we also add the connectedness via chordless æ-paths in order to reject mix. *A priori*, for classical logic it is irrelevant whether mix (which says that $A \wedge B$ implies $A \vee B$) is allowed or not since it is derivable using weakening. However, we can obtain stronger results (in particular the Decomposition Theorem 6.5.1) if we reject mix.

You have seen in Exercise 6.5.2 that Theorem 6.5.1 entails soundness and completeness of combinatorial proofs. The proof given in [Hug06a] is based on semantics, and it works equally well with our stronger criterion. In [Hug06b], Hughes gives a syntactic proof: he shows how a sequent calculus proof can be translated into a combinatorial proof, which immediately entails completeness. Then, as mentioned above, a critically chorded R&B-cograph corresponds to a proof in multiplicative linear logic [Ret03], and a skew-fibration corresponds to precisely the maps that can be constructed from contraction and weakening [Hug06b, Str07a]. This entails soundness.

Our cut elimination follows the presentation in [Str17]. An different procedure is given in [Hug06b].

Our construction of the projection in Construction 6.4.6 essentially constructs the *empire* of a proof nets via the method of [BvdW95].

The proof of Lemma 6.5.3 can be sketched as follows: First assume Φ is given. Then we can obtain f by composing the maps that are induced by the rule applications in Φ. That this is a skew fibration follows from Lemma 6.4.2. Conversely, assume f is given. Let us call a vertex in B *good* if it is in the image of f, and otherwise *bad*. Observe that whenever a vertex a in $\mathfrak{G}(B)$ is bad it cannot be connected by an edge to a good vertex. Since there is at least one good vertex, we have for every bad a a subformula $C \vee D$ in B such that (i) a is inside D, (ii) C contains a good vertex, and furthermore (iii) all vertices in D are bad. We can therefore apply $\mathsf{w}\!\downarrow$ deleting the D. Let B_0 be the formula obtained from B by repeating this process until no bad vertices remain. Then, for each atom in a define n_a be the number of vertices in $\mathfrak{G}(A)$ that f maps to a, and let B' be the formula obtained from B_0 by replacing each a by $a \vee \cdots \vee a$ where there are n_a copies. Then there is a derivation from B' to B_0 using only the $\mathsf{ac}\!\downarrow$-rule. We can define the map $f' \colon \mathfrak{G}(A) \rightarrowtail \mathfrak{G}(B')$ which takes each vertex that f maps to a to one for the new copies of a such that f is now a bijection. It is easy to see that f' is still a skew fibration. Now it follows from [Str07a, Theorem 5.1] that there is a derivation from A to B' using only m. Alternatively, this can also be shown using [Hug06b, Theorem 3.2] and the fact that a general contraction can be decomposed into $\mathsf{ac}\downarrow$ and m [BT01].

And Lemma 6.5.4 follows from the equivalence of critically chorded R&B-graphs to MLL^- proofs nets [Ret03] and the fact that $\{\mathsf{ai}\downarrow, \mathsf{ai}\uparrow, \mathsf{s}\}$ is sound and complete for MLL^- (shown in, e.g., [Ret93, Str03b, Str03a]). A direct translation from Φ into a critically chorded R&B-graph can also be obtained via Lemma 6.4.5.

7 Subatomic Proof Theory

To study normalisation procedures with some generality is very difficult: cut-elimination procedures for example are highly sensitive to variations on the form and structure of the rules of a system, where a single change in one of the rules or the addition of another warrant the need for a full new proof of cut-elimination in a new system. In this thesis we unveil a common structure behind proof systems that will allow us to generalise and understand normalisation in a simpler and more effective way. We provide a new approach within the setting of deep inference, which we call *subatomic* because we look 'inside the atoms'. It allows us to present a wide variety of propositional proof systems in such a way that *every rule* is an instance of a *single* simple linear rule scheme. We exploit this generality to study normalisation procedures and their complexity, and in particular to unveil the role played by the interactions between the rules.

In traditional Gentzen-style cut-elimination procedures cut instances are eliminated from proofs by moving upwards instances of the *mix* rule (Section 1.3). This rule conflates one instance of cut and several instances of contraction and therefore by using this technique we are in fact observing two different interactions between rules: the interactions of the cut with other rules, and the interactions of contractions with other rules. This phenomenon becomes more apparent when one considers the complexity of cut-elimination in different systems: in purely linear systems such as multiplicative linear logic the procedure does not change the size of proofs significantly, whereas as soon as contractions are introduced the size of proofs can grow exponentially or more.

In what follows we present a generalised modular normalisation theory where the different interactions between rules are dealt with separately, providing a tighter control over complexity creation. We provide a generalised procedure for cut-elimination in a generalisation of linear systems, named *splitting*. Further, we present general proof rewriting rules together with sufficient conditions for a system to be decomposable into phases containing only atomic contractions/cocontractions and a linear phase. In this way we show that this type of decomposition result holds for example for both classical logic and multiplicative additive linear logic because of shared properties in the shape of their rules. Last, we use the general reduction rules to design a local procedure to remove cycles, effectively proving the independence of decomposition and cut-elimination.

7.1 Subatomic logic

As we have mentioned in these notes, making structural rules atomic provides a surprising regularity in the inference rule schemes: it can be observed that in most deep inference systems all rules besides the atomic ones can be expressed as

$$\frac{(A \,\alpha\, B) \,\beta\, (C \,\alpha'\, D)}{(A \,\beta\, C) \,\alpha\, (B \,\beta'\, D)} \quad ,$$

where A, B, C, D are formulae and $\alpha, \beta, \alpha', \beta'$ are connectives. We call this rule shape a *medial shape*. Following this discovery, we will achieve an even greater regularity on the inference rules by looking even further, *inside the atoms*.

The main idea of subatomic logic is to consider atoms as self-dual non-commutative relations. Subatomic formulae are built by freely composing constants by connectives and

$$
\mathsf{sai}{\downarrow}\,\frac{(A \vee B)\; a\; (C \vee D)}{(A\; a\; C) \vee (B\; a\; D)} \qquad \mathsf{sai}{\uparrow}\,\frac{(A\; a\; B) \wedge (C\; a\; D)}{(A \wedge C)\; a\; (B \wedge D)}
$$

$$
{\wedge}{\downarrow}\,\frac{(A \vee B) \wedge (C \vee D)}{(A \wedge C) \vee (B \vee D)} \qquad {\wedge}{\uparrow}\,\frac{(A \vee B) \wedge (C \wedge D)}{(A \wedge C) \vee (B \wedge D)}
$$

$$
\mathsf{m}\,\frac{(A \wedge B) \vee (C \wedge D)}{(A \vee C) \wedge (B \vee D)}
$$

$$
\mathsf{sac}{\downarrow}\,\frac{(A\; a\; B) \vee (C\; a\; D)}{(A \vee C)\; a\; (B \vee D)} \qquad \mathsf{sac}{\uparrow}\,\frac{(A \wedge B)\; a\; (C \wedge D)}{(A\; a\; C) \wedge (B\; a\; D)}
$$

Figure 35: System SAKS

atoms. For example, $A \equiv ((\mathsf{f}\; a\; \mathsf{t}) \vee \mathsf{t}) \wedge (\mathsf{t}\; b\; \mathsf{f})$ is a subatomic formula for classical logic. The intuitive idea is to interpret $\mathsf{f}\, a\, \mathsf{t}$ as a positive occurrence of the atom a, and $\mathsf{t}\, a\, \mathsf{f}$ as a negative occurrence of the same atom, denoted by $\bar{a}$. We can therefore interpret A as $(a \vee \mathsf{t}) \wedge \bar{b}$. We will show an underlying structure on the shape of the inference rules, using it to present all the rules of a system as instances of a single rule scheme, including the atomic ones.

Consider for example system SKS for classical logic. We can derive the rule s from the rule

$$
\frac{(A \vee B) \wedge (C \vee D)}{(A \wedge C) \vee (B \vee D)} \quad ,
$$

which has the same 'shape' as the rule m. In fact we will show that in many systems most non-atomic rules can be made to fit this scheme as well. By using the subatomic methodology, we are able to further extend this phenomenon to atomic rules in such a way that we can present a system for classical logic where every rule of the system has the same shape. given in Figure 35.

We can view subatomic formulae as a superposition of truth values. For example, $\mathsf{f}\; a\; \mathsf{t}$ is the superposition of the two possible assignments for the atom a, and $\mathsf{t} a \mathsf{f}$ is the superposition of the possible assignments for $\bar{a}$: if we read the value on the left of the atom we assign f to a and t to $\bar{a}$, and if we read the one on the right we assign t to a and f to $\bar{a}$.

Since we consider atoms as connectives, we will give a broad definition of what connectives are, not assuming any logical characteristics or properties such as commutativity or associativity. We will therefore encompass logics with both commutative and non-commutative, associative and non-associative, dual and-self dual relations. This feature deserves to be highlighted since expressing self-dual non-commutative connectives into proof systems that enjoy cut-elimination is a challenge in Gentzen-style sequent calculi: it is impossible to have a complete analytic system with a self-dual non-commutative relation (see Section 2.3).

Definition 7.1.1. Let $\mathscr{U}$ be a denumerable set of *constants* whose elements are denoted by $u, v, w, \ldots$. Let $\mathscr{R}$ be a denumerable partially ordered set of *connectives* whose elements are denoted by α, β, γ, ... The set $\mathscr{F}$ of *subatomic formulae* (or SA formulae) contains terms defined by the grammar

$$
\mathscr{F} ::= \mathscr{U} \mid \mathscr{F}\; \mathscr{R}\; \mathscr{F} \quad .
$$

Example 7.1.2. The set $\mathscr{F}_{cl}$ of subatomic formulae for classical logic is given by the set of constants $\mathscr{U} = \{\mathsf{f}, \mathsf{t}\}$ and the set of relations $\mathscr{R} = \{\wedge, \vee\} \cup \mathscr{A}$ where $\mathscr{A}$ is a denumerable set of atoms, denoted by $a, b, \dots$ with $\mathscr{A} \cap \{\wedge, \vee\} = \varnothing$. Two examples of subatomic formulae for classical logic are

$$A \equiv ((\mathsf{f}\ a\ \mathsf{t}) \vee (\mathsf{t}\ a\ \mathsf{t})) \wedge (\mathsf{t}\ b\ \mathsf{f}) \quad \text{and} \quad B \equiv ((\mathsf{t}\ b\ \mathsf{f}) \wedge \mathsf{t}) \vee (\mathsf{f}\ a\ \mathsf{f}) \quad .$$

Example 7.1.3. The set $\mathscr{F}_{ll}$ of subatomic formulae for multiplicative linear logic is given by the set of constants $\mathscr{U} = \{\bot, 1\}$ and the set of relations $\mathscr{R} = \{\parr, \otimes\} \cup \mathscr{A}$ where $\mathscr{A}$ is a denumerable set of atoms, denoted by $a, b, \dots$ with $\mathscr{A} \cap \{\parr, \otimes\} = \varnothing$. Two examples of subatomic formulae for linear logic are

$$C \equiv ((1 \parr \bot)\ a\ 1) \otimes \bot \quad \text{and} \quad D \equiv ((\bot \parr 1)\ b\ 1) \otimes (1\ a\ \bot) \quad .$$

Given a propositional logic with certain connectives and constants, its subatomic counterpart is therefore composed of an extended language of formulae, made up from the same connectives and atoms. We can translate subatomic formulae constructed in this natural way into the 'usual' formulae by defining a simple interpretation map. Further, we can easily endow subatomic formulae with an equational theory and an involutive negation, matching that of the 'usual' formulae.

Definition 7.1.4. Let $\mathscr{G}$ be the set of formulae of a propositional logic L , and let $\mathscr{F}$ be the set of subatomic formulae with constants $\mathscr{U}$ and connectives $\mathscr{R}$. A surjective partial function $I : \mathscr{F} \to \mathscr{G}$ is called *interpretation map*. The domain of definition of I is the *set of interpretable formulae* and is denoted by $\mathscr{F}^i$. If $F \equiv I(A)$, we say that F is the *interpretation* of A, and that A is a *representation* of F.

Example 7.1.5. A natural interpretation for the set of subatomic formulae for classical logic defined in example 7.1.2 is given by considering the assignments:

$-\ I(\mathsf{t}) \equiv \mathsf{t}$;	$-\ I(\mathsf{f}) \equiv \mathsf{f}$;
$-\ \forall a \in \mathscr{A}.\ I(\mathsf{f}\ a\ \mathsf{f}) \equiv \mathsf{f}$;	$-\ \forall a \in \mathscr{A}.\ I(\mathsf{t}\ a\ \mathsf{t}) \equiv \mathsf{t}$;
$-\ \forall a \in \mathscr{A}.\ I(\mathsf{f}\ a\ \mathsf{t}) \equiv a$;	$-\ \forall a \in \mathscr{A}.\ I(\mathsf{t}\ a\ \mathsf{f}) \equiv \bar{a}$;
$-\ I(A \vee B) \equiv I(A) \vee I(B)$;	$-\ I(A \wedge B) \equiv I(A) \wedge I(B)$;

where $A, B \in \mathscr{F}^i$, and extending it in such a way that $A\,a\,B$ is interpretable iff $A = u, B = v$ with $u, v \in \{\mathsf{f}, \mathsf{t}\}$ and then $I(A\ a\ B) \equiv I(u\ a\ v)$.

For example, if $A \equiv (((\mathsf{f} \wedge \mathsf{t})\ a\ \mathsf{t}) \vee \mathsf{t}) \wedge (\mathsf{t}\ b\ \mathsf{f})$, its interpretation is $I(A) = (a \vee \mathsf{t}) \wedge \bar{b}$.

Note that the set $\mathscr{F}^i$ of interpretable formulae is composed by all formulae equal to a formula where an atom does not occur in the scope of another atom. Every other formula is not interpretable, such as $B \equiv ((\mathsf{t}\ b\ \mathsf{f}) \wedge \mathsf{t})\ a\ \mathsf{f}$.

Exercise 7.1.6. Define an interpretation for the set of subatomic formulae defined in example 7.1.3.

The useful properties of subatomic formulae become apparent when we extend the principle to atomic inference rules. Let us consider, for example, the usual contraction rule for an atom. We could obtain this rule subatomically by reading $\mathsf{f}\,a\,\mathsf{t}$ as a and $\mathsf{t}\,a\,\mathsf{f}$ as $\bar{a}$, as follows:

$$\text{we read } \frac{(\mathsf{f}\,a\,\mathsf{t}) \vee (\mathsf{f}\,a\,\mathsf{t})}{(\mathsf{f} \vee \mathsf{f})\,a\,(\mathsf{t} \vee \mathsf{t})} \text{ as } \frac{a \vee a}{a} \text{ and we read } \frac{(\mathsf{t}\,a\,\mathsf{f}) \vee (\mathsf{t}\,a\,\mathsf{f})}{(\mathsf{t} \vee \mathsf{t})\,a\,(\mathsf{t} \vee \mathsf{t})} \text{ as } \frac{\bar{a} \vee \bar{a}}{\bar{a}}.$$

These rules are therefore generated by the linear scheme

$$\frac{(A\,a\,B) \vee (C\,a\,D)}{(A \vee C)\,a\,(B \vee D)}, \text{ where } A, B, C, D \text{ are formulae.}$$

The non-linearity of the contraction rule has been pushed from the atoms to the units.

Similarly, we can consider the atomic identity rule. It can be obtained subatomically as follows:

$$\text{we read } \frac{(\mathsf{f}\,a\,\mathsf{t}) \vee (\mathsf{t}\,a\,\mathsf{f})}{(\mathsf{f} \vee \mathsf{t})\,a\,(\mathsf{t} \vee \mathsf{f})} \text{ as } \frac{\mathsf{t}}{a \vee \bar{a}}.$$

Similarly to the contraction rule, it is generated by the linear scheme

$$\frac{(A \vee B)\,a\,(C \vee D)}{(A\,a\,C) \vee (B\,a\,D)}, \text{ where } A, B, C, D \text{ are formulae.}$$

It is quite plain to see that both the subatomic contraction rule and the subatomic introduction rule have the same *medial* shape, typical of logical rules in deep inference. We have therefore uncovered an underlying structure behind the shape of inference rules, that we will exploit to obtain a general characterisation of rules.

To make use of the general characterisation, we will impose some restrictions on $\alpha, \nu, \beta, \gamma$. These conditions strike a balance between being general enough to encompass a wide variety of logics and being explicit enough to enable us to generalise procedure such as cut-elimination and decomposition. To do so, we exploit the dualities present in the inference rules, and we introduce a notion of polarity in the pairs of dual relations. The idea behind it is rather to assign which of the relations in the pair is 'stronger' than the other. Intuitively, it loosely corresponds to assigning which relation of the pair will imply the other. For example, in classical logic $A \wedge B$ implies $A \vee B$, and thus we will assign $\wedge$ to be *strong* and $\vee$ to be *weak*.

Definition 7.1.7. For each pair of connectives $\{\alpha, \overline{\alpha}\}$, we give a polarity assignment: we call one connective of the pair *strong* and the other one *weak*.

If α is strong and $\overline{\alpha}$ is weak, we will write $\alpha^M = \overline{\alpha}^M = \alpha$ and $\alpha^m = \overline{\alpha}^m = \overline{\alpha}$. Self-dual connectives are both strong and weak.

Definition 7.1.8. A *subatomic proof system* SA with set of formulae $\mathscr{F}$ is

- a collection of inference rules of the shape $\dfrac{(A\,\beta\,B)\,\alpha\,(C\,\beta\,D)}{(A\,\alpha\,C)\,\beta\,(B\,\alpha^m\,D)}$, $\alpha, \beta \in \mathscr{R}$, called *down-rules*,
- a collection of inference rules of the shape $\dfrac{(A\,\beta\,B)\,\alpha\,(C\,\beta^M\,D)}{(A\,\alpha\,C)\,\beta\,(B\,\alpha\,D)}$, $\alpha, \beta \in \mathscr{R}$, called *up-rules*,

$$\mathsf{sai}\downarrow \frac{(A \parr B)\, a\, (C \parr D)}{(A\, a\, C) \parr (B\, a\, D)} \qquad \mathsf{sai}\uparrow \frac{(A\, a\, B) \otimes (C\, a\, D)}{(A \otimes C)\, a\, (B \otimes D)}$$

$$\otimes\downarrow \frac{(A \parr B) \otimes (C \parr D)}{(A \otimes C) \parr (B \parr D)} \qquad \otimes\uparrow \frac{(A \parr B) \otimes (C \otimes D)}{(A \otimes C) \parr (B \otimes D)}$$

Figure 36: System SAMLLS

- a collection of rules $= \frac{A}{B}$ and $= \frac{\overline{A}}{\overline{B}}$, for every axiom $A = B$ of the equational theory $=$ on $\mathscr{F}$, called *equality rules*.

Example 7.1.9. We consider $\wedge$ as strong and $\vee$ as weak in classical logic. The subatomic proof system SAKS is given by the inference rules in Figure 35, together with the equality rules given by $= \frac{A}{B}$ for every equality in the equational theory for classical logic formulae.

Example 7.1.10. We consider $\otimes$ as strong and $\parr$ as weak in multiplicative linear logic. The subatomic proof system SAMLLS is given by the inference rules in Figure 36 together with the equality rules given by the equational theory of MLL.

We can straightforwardly build deep inference derivations as is usual in the literature, by vertical composition through an inference rule and horizontal composition by logical relations, and the interpretation map is easily extended from formulae to derivations. The notion of proof is generalised as well.

Definition 7.1.11. Let $1 \in \mathscr{U}$ be a distinguished constant. A *proof* of A is a derivation Π whose premiss is 1. We denote proofs by $\begin{array}{c}\Pi \Big\| \\ A\end{array}$.

7.2 Splitting

There are many different cut-elimination techniques in the deep inference literature, exploiting different aspects of the proof systems they work on. In this assortment, a particular methodology does however stand out for its generality: cut-elimination via *splitting* (see Section 4.2. The generality of this procedure points towards the fact that it exploits some properties that are common to all these systems.

Splitting is based on a simple idea: to show that an atomic cut involving a and $\bar{a}$ is admissible, we trace a and $\bar{a}$ to the top of the proof to find two independent subproofs, the premiss of one containing the dual of a and the other one containing the dual of $\bar{a}$. In this way we obtain two independent 'pieces' that we can rearrange to get a new cut-free proof.

Proofs of cut-elimination by splitting therefore rely on two main properties of a proof system: the *dualities* present in it to ensure that each of the independent subproofs contains the dual of an atom involved in the cut, and the *shape* of the linear rules ensuring that the two proofs remain independent above the cut. It is precisely a formal characterisation of these properties that we will provide, enabling us to understand why they are enough to guarantee cut-elimination.

As we saw in Section 4.2, to trace a connective through the proof from the bottom to the top, we need its scope to widen. Accordingly, we will consider systems where the shape of the rules ensures the widening of the scope. In what follows, we will characterise *splittable systems*, *i.e.*, systems with sufficient conditions to ensure cut-elimination through a splitting procedure.

Definition 7.2.1. A system $\mathsf{SA}^{\downarrow}$ is *splittable* if:

1. There is a distinguished associative and commutative strong connective $\times$ with unit 1 and dual $+$ with unit 0,
2. SA is uniquely composed of down-rules of the form

$$\alpha\!\downarrow \frac{(A+B)\,\alpha\,(C+D)}{(A\,\alpha\,C)+(B\,\alpha^m\,D)} \quad ,$$

   for every connective $\alpha \in \mathscr{R}$.

3. The equality $u + \bar{u} = 1$ holds for every unit $u \in \mathscr{U}$,
4. The equality $1\,\alpha^M\,1 = 1$ holds for every $\alpha \in \mathscr{R}$.

Example 7.2.2. $\mathsf{SAMLLS}^{\downarrow}$ is splittable, and the distinguished connective $+$ introducing dualities is $\parr$.

A difficulty of splitting is finding the right induction measure for every system. In the literature, each splitting theorem for each proof system uses a different induction measure tailored specifically for it. By providing a general splitting theorem, we not only give a formal definition of what a splitting theorem is, but also give a one-size-fits-all induction measure based on the length of the proof that works for every splittable system, taking the search for an induction measure out of the process for designing a proof system. The following theorems hold for splittable systems.

Definition 7.2.3. Given a derivation ϕ, we define the *length* of ϕ as the number of rules in ϕ different from the equality rules for the associativity and commutativity of $+$, the unit rule for $+$ and the unit assignments for $+$. We denote it by $|\phi|_+$.

Theorem 7.2.4 (Shallow Splitting). *If $\mathsf{SA}^{\downarrow}$ is splittable, for every formulae A, B, C, for every connective $\alpha \neq +$, for every proof*

$$\begin{array}{c} \Pi \Big\| \mathsf{SA}^{\downarrow} \\ (A\,\alpha\,B)+C \end{array}$$

there exist formulae Q_1, Q_2 and derivations

$$\begin{array}{c} Q_1\,\overline{\alpha}\,Q_2 \\ \Phi \Big\| \mathsf{SA}^{\downarrow} \\ C \end{array} \quad , \quad \begin{array}{c} \Pi_1 \Big\| \mathsf{SA}^{\downarrow} \\ A+Q_1 \end{array} \quad \text{and} \quad \begin{array}{c} \Pi_2 \Big\| \mathsf{SA}^{\downarrow} \\ B+Q_2 \end{array} \quad .$$

with $|\Pi_1|_+ + |\Pi_2|_+ \leq |\Pi|_+$.

Theorem 7.2.5 (Context Reduction). *Let* $\mathsf{SA}^{\downarrow}$ *be a splittable system. For any formula* A *and for any context* $S\{\ \}$*, given a proof* $\begin{array}{c}\Pi \Vert \mathsf{SA}^{\downarrow} \\ S\{A\}\end{array}$*, there exist a formula* K*, a provable context* $H\{\ \}$ *and derivations*

$$\begin{array}{c}\Theta \Vert \mathsf{SA}^{\downarrow} \\ A + K\end{array} \qquad \textit{and} \qquad \begin{array}{c} H\{\{\ \} + K\} \\ \chi \Vert \mathsf{SA}^{\downarrow} \\ S\{\ \}\end{array} \qquad .$$

We can easily see that the theorems above are generalisations of those we presented in Section 4.2. They can be proved with the same proof scheme. However, for this result to straightforwardly hold in the original non-subatomic systems, we will need to show that the cut-free proofs obtained from proofs of the non-subatomic original system via this procedure are interpretable themselves, and therefore correspond to proofs in the original system. For that, we will pay particular attention to tame proofs, in which no inference rule occurs in the scope of an atom. If the interpretation I is built in a natural way, every proof of the original system will be represented by a tame proof in SA. The interpretability of tame proofs is preserved by splitting as long as interpretability is preserved by duals. In that case, as a corollary, interpretability will be preserved by splitting.

Definition 7.2.6. We say that a system SA with a natural interpretation I, negation $\bar{\cdot}$ and an equational theory $=$ is *preservable* when:

1. If A is interpretable and $A =_{+} B$, then B is interpretable ;
2. If $A \,\alpha\, B$ is interpretable, $\alpha \in \mathscr{R}$, then A and B are interpretable ;
3. If $A \,a\, B$ is interpretable and $A + A' = 1$, $B + B' = 1$ then $A' \,a\, B'$ is interpretable for $a \in \mathscr{A}$;
4. If A is interpretable, then $\overline{A}$ is interpretable ;
5. The atoms of $\mathscr{A}$ are non-commutative, non-associative and non-unitary.

We can now add the following details about the preservation of interpretability:

Lemma 7.2.7. *Let* $SA^{\downarrow}$ *be preservable and*

$$\begin{array}{c}\Pi \Vert \mathsf{SA}^{\downarrow} \\ (A \,\alpha\, B) + C\end{array}$$

be a tame proof. Then the derivations Π_1, Π_2 *and* Φ *obtained from Theorem 7.2.4 are tame. Furthermore, if* α *is an atom then* Π_1 *and* Π_2 *are equalities.*

Lemma 7.2.8. *Let* $SA^{\downarrow}$ *be preservable and* $\begin{array}{c}\Pi \Vert \mathsf{SA}^{\downarrow} \\ S\{A\}\end{array}$ *be a tame proof. Then the derivation* Θ *obtained from Theorem 7.2.5 is tame.*

Furthermore, if $\{\ \}$ *is not in the scope of an atom in* $S\{\ \}$ *and* Π *is tame, then the derivation* χ *obtained from Theorem 7.2.5 is tame.*

The proof scheme is identical to the one we showed in Section 4.2. Here, we will only sketch the proofs.

Lemma 7.2.9. *If* $\mathsf{SA}^{\downarrow}$ *is splittable, then for every proof*

$$\begin{array}{c} \Pi \Big\| \mathsf{SA}^{\downarrow} \\ u + C \end{array}$$

where $u \in \mathscr{U}$*, there is a derivation*

$$\begin{array}{c} \bar{u} \\ \Phi \Big\| \mathsf{SA}^{\downarrow} \\ C \end{array} \quad .$$

Furthermore, if $\mathsf{SA}^{\downarrow}$ *is preservable, then if* Π *is tame we have that* Φ *is tame.*

Proof. We take

$$\Phi \equiv \times\!\downarrow \frac{(\bar{u}+0) \times \begin{array}{c} \Pi \Big\| \\ u + C \end{array}}{= \dfrac{\bar{u} \times u}{0} + 0 + C} \quad .$$

□

Sketch of proof of Theorem 7.2.4 and Lemma 7.2.7. We define $=_+$ as the equivalence relation on formulae defined by the axioms for the associativity, commutativity, unit of $+$ and constant assignments for $+$. We will proceed by induction on $|\Pi|_+$.

If $|\Pi|_+ = 1$, then $A =_+ v, B =_+ w$ and $v \, \alpha \, w =_+ u$, with $u + C =_+ 1$. By Lemma 7.2.9, there is a derivation $\begin{array}{c} \bar{u} \\ \Phi' \Big\| \mathsf{SA}^{\downarrow} \\ C \end{array}$ and we take:

$$\Phi \equiv \begin{array}{c} = \dfrac{\bar{v} \, \bar{\alpha} \, \bar{w}}{\bar{u}} \\ \Phi' \Big\| \\ C \end{array} \quad , \qquad \Pi_1 \equiv \; =_+ \frac{1}{=_+ \dfrac{v}{A} + \bar{v}} \quad \text{and} \quad \Pi_2 \equiv \; =_+ \frac{1}{=_+ \dfrac{w}{B} + \bar{w}} \quad .$$

Φ' is tame and $\bar{v} \, \bar{\alpha} \, \bar{w}$ is interpretable, and therefore Φ is tame. Furthermore, Π_1 and Π_2 are tame and equalities.

If $|\Pi|_+ > 1$, we prove the inductive step for all the possible cases of the bottom inference rule ρ of ϕ.

Inspection of the rules provides us with the following possible cases:

$$(1)\quad \Pi =_{+} {}_{\rho}\dfrac{\begin{array}{c}\Pi' \,\|\, \mathsf{SA}^{\downarrow} \\ (A\,\alpha\,B) + C'\end{array}}{(A\,\alpha\,B) + C} \qquad ;$$

$$(2)\quad \Pi =_{+} {}_{\times\downarrow}\dfrac{\begin{array}{c}\Pi' \,\|\, \mathsf{SA}^{\downarrow} \\ (((A\,\alpha\,B) + C_1) \times (C_2 + C_3)) + C_4\end{array}}{(A\,\alpha\,B) + C_2 + (C_1 \times C_3) + C_4} \qquad ;$$

$$(3)\quad \Pi =_{+} {}_{=}\dfrac{\begin{array}{c}\Pi' \,\|\, \mathsf{SA}^{\downarrow} \\ (((A\,\alpha\,B) + C_1)\,\beta\,u_\beta) + C_2\end{array}}{(A\,\alpha\,B) + C_1 + C_2} \qquad ;$$

$$(4)\quad \Pi =_{+} {}_{=}\dfrac{\begin{array}{c}\Pi' \,\|\, \mathsf{SA}^{\downarrow} \\ (u_\beta\,\beta\,((A\,\alpha\,B) + C_1)) + C_2\end{array}}{(A\,\alpha\,B) + C_1 + C_2} \qquad ;$$

$$(5)\quad \Pi =_{+} {}_{\rho}\dfrac{\begin{array}{c}\Pi' \,\|\, \mathsf{SA}^{\downarrow} \\ (A'\,\alpha\,B) + C\end{array}}{(A\,\alpha\,B) + C} \qquad ;$$

$$(6)\quad \Pi =_{+} {}_{\rho}\dfrac{\begin{array}{c}\Pi' \,\|\, \mathsf{SA}^{\downarrow} \\ (A\,\alpha\,B') + C\end{array}}{(A\,\alpha\,B) + C} \qquad ;$$

$$(7)\quad \Pi =_{+} {}_{\alpha\downarrow}\dfrac{\begin{array}{c}\Pi' \,\|\, \mathsf{SA}^{\downarrow} \\ ((A + C_1)\,\alpha\,(B + C_2)) + C_3\end{array}}{((A\,\alpha\,B) + (C_1\,\overline{\alpha}\,C_2)) + C_3} \qquad \text{if } \alpha \text{ is strong ;}$$

$$(8)\quad \Pi =_{+} {}_{\alpha\downarrow}\dfrac{\begin{array}{c}\Pi' \,\|\, \mathsf{SA}^{\downarrow} \\ ((A + C_1)\,\overline{\alpha}\,(B + C_2)) + C_3\end{array}}{((A\,\alpha\,B) + (C_1\,\overline{\alpha}\,C_2)) + C_3} \qquad \text{if } \alpha \text{ is weak ;}$$

$$(9)\quad \Pi =_{+} {}_{\alpha\downarrow}\dfrac{\begin{array}{c}\Pi' \,\|\, \mathsf{SA}^{\downarrow} \\ ((A + C_1)\,\alpha\,(B + C_2)) + C_3\end{array}}{((A\,\alpha\,B) + (C_1\,\alpha\,C_2)) + C_3} \qquad \text{if } \alpha \text{ is weak ;}$$

$$(10)\quad \Pi =_{+} {}_{=}\dfrac{\begin{array}{c}\Pi' \,\|\, \mathsf{SA}^{\downarrow} \\ (B\,\alpha\,A) + C\end{array}}{(A\,\alpha\,B) + C} \qquad \text{if } \alpha \text{ is commutative ;}$$

$$(11)\quad \Pi =_{+} \;\; \genfrac{}{}{0pt}{}{\Pi' \Big\| \mathsf{SA}^{\downarrow}}{= \dfrac{((A \,\alpha\, B_1) \,\alpha\, B_2) + C}{(A \,\alpha\, (B_1 \,\alpha\, B_2)) + C}} \qquad \text{if } \alpha \text{ is associative ;}$$

$$(12)\quad \Pi =_{+} \;\; \genfrac{}{}{0pt}{}{\Pi' \Big\| \mathsf{SA}^{\downarrow}}{= \dfrac{(A_1 \,\alpha\, (A_2 \,\alpha\, B)) + C}{((A_1 \,\alpha\, A_2) \,\alpha\, B) + C}} \qquad \text{if } \alpha \text{ is associative ;}$$

$$(13)\quad \Pi =_{+} \;\; \genfrac{}{}{0pt}{}{\Pi' \Big\| \mathsf{SA}^{\downarrow}}{= \dfrac{A + C}{(A \,\alpha\, u_\alpha) + C}} \qquad \text{if } \alpha \text{ is unitary, with } B =_{+} u_\beta \text{ ;}$$

$$(14)\quad \Pi =_{+} \;\; \genfrac{}{}{0pt}{}{\Pi' \Big\| \mathsf{SA}^{\downarrow}}{= \dfrac{B + C}{(u_\alpha \,\alpha\, B) + C}} \qquad \text{if } \alpha \text{ is unitary, with } A =_{+} u_\beta \text{ ;}$$

$$(15)\quad \Pi =_{+} \;\; \genfrac{}{}{0pt}{}{\Pi' \Big\| \mathsf{SA}^{\downarrow}}{= \dfrac{u + C}{(v \,\alpha\, w) + C}} \qquad \text{with } A =_{+} v \text{ and } B =_{+} w \text{ .}$$

We will describe only a few of the induction cases.

(1) There are derivations

$$\Phi =_{+} \;\; \genfrac{}{}{0pt}{}{Q_1 \,\overline{\alpha}\, Q_2}{\genfrac{}{}{0pt}{}{\Phi' \Big\| \mathsf{SA}^{\downarrow}}{\rho\,\dfrac{C'}{C}}} \quad , \qquad \genfrac{}{}{0pt}{}{\Pi_1 \Big\| \mathsf{SA}^{\downarrow}}{A + Q_1} \qquad \text{and} \qquad \genfrac{}{}{0pt}{}{\Pi_2 \Big\| \mathsf{SA}^{\downarrow}}{B + Q_2} \quad ,$$

with $\mathsf{size}(\Pi_1) + \mathsf{size}(\Pi_2) \leq \mathsf{size}(\Pi') < \mathsf{size}(\Pi)$.

If Π is tame, then ρ and Π_1, Π_2 and Π' are tame. Hence Φ is tame.

Furthermore, if α is an atom then by the induction hypothesis Π_1 and Π_2 are equalities.

(7) There are derivations

$$\genfrac{}{}{0pt}{}{H_1 \,\overline{\alpha}\, H_2}{\genfrac{}{}{0pt}{}{\Phi' \Big\| \mathsf{SA}^{\downarrow}}{C_3}} \quad , \qquad \genfrac{}{}{0pt}{}{\Pi_1 \Big\| \mathsf{SA}^{\downarrow}}{A + C_1 + H_1} \qquad \text{and} \qquad \genfrac{}{}{0pt}{}{\Pi_2 \Big\| \mathsf{SA}^{\downarrow}}{B + C_2 + H_2} \quad ,$$

with $\mathsf{size}(\Pi_1) + \mathsf{size}(\Pi_2) \leq \mathsf{size}(\Pi') < \mathsf{size}(\Pi)$.

We take $Q_1 \equiv C_1 + H_1$, $Q_2 \equiv C_2 + H_2$ and

$$\Phi =_+ \quad \cfrac{\overline{\alpha}\downarrow \dfrac{(C_1 + H_1)\ \overline{\alpha}\ (C_2 + H_2)}{}}{(C_1\ \overline{\alpha}\ C_2) + \begin{array}{c} H_1\ \overline{\alpha}\ H_2 \\ {\scriptstyle \Phi'}\Big\| \\ C_3 \end{array}} \quad .$$

If Π is tame, then Π' is tame and by induction hypothesis Π_1, Π_2 and Φ' are tame.

If α is an atom, then by the induction hypothesis Π_1 and Π_2 are equalities. Then $(C_1 + H_1)\ \overline{\alpha}\ (C_2 + H_2)$ is interpretable by condition 3 of preservability. Therefore, Φ is tame.

If Π is tame and α is not an atom, then Φ is trivially tame since C_1, H_1, C_2, H_2 are interpretable and Φ' is tame.

(9) There are derivations

$$\begin{array}{c} H_1\ \overline{\alpha}\ H_2 \\ {\scriptstyle \Phi'}\Big\|{\scriptstyle \mathsf{SA}\downarrow} \\ F_3 \end{array} \quad , \quad \begin{array}{c} {\scriptstyle \Pi_1}\Big\|{\scriptstyle \mathsf{SA}\downarrow} \\ A + C_1 + H_1 \end{array} \quad \text{and} \quad \begin{array}{c} {\scriptstyle \Pi_2}\Big\|{\scriptstyle \mathsf{SA}\downarrow} \\ B + C_2 + H_2 \end{array} \quad ,$$

with $\mathsf{size}(\Pi_1) + \mathsf{size}(\Pi_2) \leq \mathsf{size}(\Pi') < \mathsf{size}(\Pi)$.

We take $Q_1 \equiv C_1 + H_1$, $Q_2 \equiv C_2 + H_2$ and

$$\Phi =_+ \quad \cfrac{\alpha^m\downarrow \dfrac{(C_1 + H_1)\ \overline{\alpha}\ (C_2 + H_2)}{}}{(C_1\ \alpha\ C_2) + \begin{array}{c} H_1\ \overline{\alpha}\ H_2 \\ {\scriptstyle \Phi'}\Big\| \\ C_3 \end{array}} \quad .$$

If Π is tame, then Π' is tame and by induction hypothesis Π_1, Π_2 and Φ' are tame.

If α is an atom, then by the induction hypothesis Π_1 and Π_2 are equalities. Then $(C_1 + H_1)\ \overline{\alpha}\ (C_2 + H_2)$ is interpretable by condition 3 of preservability. Therefore, Φ is tame.

If Π is tame and α is not an atom, then Φ is trivially tame since C_1, H_1, C_2, H_2 are interpretable and Φ' is tame.

□

Exercise 7.2.10. Finish the missing induction cases.

Proof of Theorem 7.2.5 and Lemma 7.2.8. We proceed by induction on the number of relations $\alpha \neq +$ that $\{\ \}$ is in the scope of in $S\{\ \}$. We denote it by $|S|_+$.

If $|S|_+ = 0$, then $S\{A\} =_+ A + K$ and we take $\Theta =_+ \phi$ and $H\{\ \} = \{\ \}$.

If $S\{A\} =_+ (S'\{A\}\ \beta\ B) + C$ with $\beta \neq +$, we apply Theorem 7.2.4 to Π. There exist derivations

$$\begin{array}{c} Q_1\ \overline{\beta}\ Q_2 \\ \Phi \Big\Vert \mathsf{SA}^{\downarrow} \\ C \end{array} \quad , \quad \begin{array}{c} \Pi_1 \Big\Vert \mathsf{SA}^{\downarrow} \\ S'\{A\} + Q_1 \end{array} \quad \text{and} \quad \begin{array}{c} \Pi_2 \Big\Vert \mathsf{SA}^{\downarrow} \\ B + Q_2 \end{array}$$

such that Π_1, Π_2 and Φ are tame if Π is tame.

We apply the induction hypothesis to Π_1 since $|S'|_+ < |S|_+$. There are derivations

$$\begin{array}{c} \Theta \Big\Vert \mathsf{SA}^{\downarrow} \\ A + K \end{array} \quad , \quad \begin{array}{c} H'\{\{\ \} + K\} \\ \chi' \Big\Vert \mathsf{SA}^{\downarrow} \\ S'\{\ \} + Q_1 \end{array} \quad ,$$

with H' a provable context, such that Θ is tame if ϕ_1 is tame.

We take $H\{\ \} = H'\{\ \}\ \beta^M\ 1$. We have $H\{1\} = H'\{1\}\ \beta^M\ 1 = 1\ \beta^M\ 1 = 1$, and we can build in $\mathsf{SA}^{\downarrow}$

$$\chi \equiv \beta{\downarrow}\frac{\boxed{\begin{array}{c} H'\{\{\ \} + K\} \\ \chi' \Big\Vert \\ S'\{\ \} + Q_1 \end{array}}\ \beta^M\ \boxed{\begin{array}{c} \Pi_2 \Big\Vert \\ B + Q_2 \end{array}}}{S'\{\ \}\ \beta\ B + \boxed{\begin{array}{c} Q_1\ \overline{\beta}\ Q_2 \\ \Phi \Big\Vert \\ C \end{array}}} \quad .$$

If $\{\ \}$ is not in the scope of an atom in $S\{\ \}$ and Π is tame, then by the induction hypothesis χ' is tame and $\{\ \}$ is not in the scope of an atom in $H'\{\ \}$. Since β is not an atom, $\{\ \}$ is not in the scope of an atom in $H\{\ \}$ and χ is tame.

We proceed likewise if $S\{A\} =_+ (B\ \beta\ S'\{A\}) + C$.

□

As a corollary of shallow splitting and context reduction we can show the admissibility of a class of up-rules. The main idea is that through splitting we can separate a proof into "building blocks" that are independently provable. We can then easily combine these building blocks differently to obtain a new proof with the same conclusion.

Since tameness is preserved by splitting, cut-free proofs obtained from tame proofs will be tame themselves. The cut-free proofs obtained from non-subatomic proofs will therefore be interpretable, and we can ensure that this cut-elimination result corresponds to cut-elimination in the original system.

When designing a proof system that enjoys cut-elimination, we will therefore only have to ensure that the interpretation map is preservable. This is quite an easy task, since the conditions for an interpretation map to be natural are very lenient, and therefore there is much freedom to design an interpretation to suit many needs.

Definition 7.2.11. Rules of the form $\alpha\!\uparrow \dfrac{(A\,\alpha\,B)\times(C\,\alpha^M\,D)}{(A\times C)\,\alpha\,(B\times D)}$ are *cuts.*

Corollary 7.2.12 (Admissibility of cuts)**.** *Let* SA *be a splittable proof system.*

For any formulae A, B, C, D*, any context* S*, any connective* $\alpha \neq +$*, given a proof*

$$\Pi \equiv \begin{array}{c} \Pi' \big\Vert \mathsf{SA}{\downarrow} \\ S\left\{ \alpha\!\uparrow \dfrac{(A\,\alpha\,B)\times(C\,\alpha^M\,D)}{(A\times C)\,\alpha\,(B\times D)} \right\} \end{array} \quad ,$$

there is a proof

$$\begin{array}{c} \Psi \big\Vert \mathsf{SA}{\downarrow} \\ S\{(A\times C)\,\alpha\,(B\times D)\} \end{array} \quad ,$$

i.e., cuts are admissible.

Furthermore, if Π *is tame and* α *is not an atom,* Ψ *is tame.*

Proof. We apply Theorem 7.2.5 to Π.

There are derivations

$$\begin{array}{c} \Theta \big\Vert \mathsf{SA}{\downarrow} \\ ((A\,\alpha\,B)\times(C\,\alpha^M\,D)) + K \end{array} \quad \text{and} \quad \begin{array}{c} H\{\{\ \} + K\} \\ \chi \big\Vert \mathsf{SA}{\downarrow} \\ S\{\ \} \end{array} \quad ,$$

with $H\{1\} = 1$.

We apply Theorem 7.2.4 to Θ. There exist derivations

$$\begin{array}{c} Q_1 + Q_2 \\ \Phi \big\Vert \mathsf{SA}{\downarrow} \\ K \end{array} \quad , \quad \begin{array}{c} \Pi_1 \big\Vert \mathsf{SA}{\downarrow} \\ (A\,\alpha\,B) + Q_1 \end{array} \quad \text{and} \quad \begin{array}{c} \Pi_2 \big\Vert \mathsf{SA}{\downarrow} \\ (C\,\alpha^M\,D) + Q_2 \end{array} \quad .$$

We apply Theorem 7.2.4 to Π_1 and Π_2 and we obtain

$$\begin{array}{c} Q_A\,\overline{\alpha}\,Q_B \\ \Phi_1 \big\Vert \mathsf{SA}{\downarrow} \\ Q_1 \end{array} \quad , \quad \begin{array}{c} \Pi_3 \big\Vert \mathsf{SA}{\downarrow} \\ Q_A + A \end{array} \quad \text{and} \quad \begin{array}{c} \Pi_4 \big\Vert \mathsf{SA}{\downarrow} \\ Q_B + B \end{array} \quad ,$$

$$\begin{array}{c} Q_C\,\alpha^m\,Q_D \\ \Phi_2 \big\Vert \mathsf{SA}{\downarrow} \\ Q_2 \end{array} \quad , \quad \begin{array}{c} \Pi_5 \big\Vert \mathsf{SA}{\downarrow} \\ Q_C + C \end{array} \quad \text{and} \quad \begin{array}{c} \Pi_6 \big\Vert \mathsf{SA}{\downarrow} \\ Q_D + D \end{array} \quad .$$

We can then build the following proof in $\mathsf{SA}^{\downarrow}$

$$
\Psi = \begin{array}{c}
H\left\{\begin{array}{c}
\alpha^M{\downarrow}\,\dfrac{\boxed{\times{\downarrow}\,\dfrac{\boxed{\begin{array}{c}\Pi_3\,\Vert\\ A+Q_A\end{array}} \times \boxed{\begin{array}{c}\Pi_5\,\Vert\\ C+Q_C\end{array}}}{(A\times C)+Q_A+Q_C}} \;\alpha^M\; \boxed{\times{\downarrow}\,\dfrac{\boxed{\begin{array}{c}\Pi_4\,\Vert\\ B+Q_B\end{array}} \times \boxed{\begin{array}{c}\Pi_6\,\Vert\\ D+Q_D\end{array}}}{(B\times D)+Q_B+Q_D}}}{((A\times C)\,\alpha\,(B\times D)) + \boxed{\begin{array}{c}\overline{\alpha}{\downarrow}\,\dfrac{(Q_A+Q_C)\;\overline{\alpha}\;(Q_B+Q_D)}{\boxed{\begin{array}{c}Q_A\;\overline{\alpha}\;Q_B\\ \Phi_1\,\Vert\\ Q_1\end{array}} + \boxed{\begin{array}{c}Q_C\;\alpha^m\;Q_D\\ \Phi_2\,\Vert\\ Q_2\end{array}}}\\ \Phi\,\Vert\\ K\end{array}}}
\end{array}\right\} \\
\chi\,\Vert \\
S\{(A\times C)\,\alpha\,(B\times D)\}
\end{array}
\quad .
$$

If Π is tame, then $\{\ \}$ is not in the scope of an atom in $S\{\ \}$ and $\Pi_3, \Pi_4, \Pi_5, \Pi_6, \Phi_1, \Phi_2$ and χ are tame. Therefore, if α is not an atom, Ψ is tame.

□

We have shown that splitting hinges only on the shape of rules and on dualities. The general splitting methodology is very robust: it is based on properties that are present in systems with very different expressiveness and therefore it can be expanded to include an extremely wide variety of relations as long as they are introduced by rules of non-contractive shape.

7.3 Decomposition

Splitting allows us to understand the interactions of the cut with splittable linear rules, but how about contractions? It is known that we can decompose classical logic and multiplicative additive linear logic proofs into a linear phase and a phase made-up only of contractions via decomposition results. We study this phenomenon, providing general rewriting rules that encompass the reductions presented in both systems. We thus show that both decomposition results are a consequence of precisely the same properties.

Additionally, it has long been conjectured that it is possible to achieve a further decomposition of these systems, permuting not only the atomic contraction but a whole family of *contractive* rules towards the bottom of a derivation. The generalised rewriting rules that we present allow us to permute medial rules with linear rules, including cuts. The regularity provided by subatomic systems is a big simplification for the study of these interactions: by having a single shape we only have to consider two non-trivial permutation cases.

The first step in the generalisation is to characterise the family of rules that will be permuted. Unsurprisingly, the rules that we will be able to permute downwards/upwards in a derivation correspond to the rules involved in making contraction atomic. We will call them *contractions* as well.

We define *ν-contractive systems*, which correspond to those systems where we can recover *general contractions*

$$\frac{A \,\nu\, A}{A} \quad .$$

Definition 7.3.1. Let ν be a distinguished relation with unit $\triangledown$, and $\overline{\nu}$ its dual with unit $\vartriangle$. A *ν-contractive system* SA is a subatomic proof system where:

- For every relation α there is a down rule of the form

$$\alpha c\,\frac{(A \,\alpha\, B) \,\nu\, (C \,\alpha\, D)}{(A \,\nu\, C) \,\alpha\, (B \,\nu\, D)} \quad ,$$

  that we call *contraction for α* and its dual up rule that we call *cocontraction for $\overline{\alpha}$*.

- For every constant $u \in \mathscr{U}$ there are equalities of the form $u \,\nu\, u = u$ and $\bar{u} \,\overline{\nu}\, \bar{u} = \bar{u}$. We call the equality rules $=\dfrac{u \,\nu\, u}{u}$ the *contraction equality rule for u* and $=\dfrac{\bar{u}}{\bar{u} \,\overline{\nu}\, \bar{u}}$ the *cocontraction equality rule for $\bar{u}$*.

- For every constant $u \in \mathscr{U}$, $w\,\dfrac{\triangledown}{u}$ and its dual $\bar{w}\,\dfrac{\bar{u}}{\vartriangle}$ are derivable in SA. We call these unitary instances of (co)contraction rules *weakening* and *coweakening* respectively.

- For every relation α there are equalities $\triangledown \,\alpha\, \triangledown = \triangledown$ and $\vartriangle \,\alpha\, \vartriangle = \vartriangle$.

Complexity is created by the duplication of atomic contractions when they are permuted through other rules. It is our goal to understand this phenomenon in the best possible generality, i.e. to keep track of the creation and duplication of atoms when any contraction rule is permuted downwards in a derivation. However, when permuting contraction rules we may create an unbounded number of other contractive and cocontractive rules.

By observing the subatomic form of known rewriting rules that permute atomic contractions downwards in derivations, a novel way of controlling this phenomenon arises: we will show that it is possible to move 'blocks' of nested contraction rules together, in such a way that we are no longer concerned by the number of (co)contraction rules created by the procedure.

Consider this reduction, corresponding to permuting an atomic contraction through an atomic cut:

$$a{\uparrow}\,\frac{\boxed{ac\,\dfrac{(\mathsf{f}\,a\,\mathsf{t}) \vee (\mathsf{f}\,a\,\mathsf{t})}{\boxed{\dfrac{\mathsf{f} \vee \mathsf{f}}{\mathsf{f}}}\; a \;\boxed{\dfrac{\mathsf{t} \vee \mathsf{t}}{\mathsf{t}}}}} \wedge (\mathsf{t}\,a\,\mathsf{f})}{(\mathsf{f} \wedge \mathsf{t})\,a\,(\mathsf{t} \wedge \mathsf{f})} \quad \longrightarrow \quad \wedge{\uparrow}\,\frac{((\mathsf{f}\,a\,\mathsf{t}) \vee (\mathsf{f}\,a\,\mathsf{t})) \wedge \boxed{a\bar{c}\,\dfrac{\boxed{\dfrac{\mathsf{t}}{\mathsf{t} \wedge \mathsf{t}}}\; a \;\boxed{\dfrac{\mathsf{f}}{\mathsf{f} \wedge \mathsf{f}}}}{(\mathsf{t}\,a\,\mathsf{f}) \wedge (\mathsf{t}\,a\,\mathsf{f})}}}{\boxed{ac\,\dfrac{\boxed{a{\uparrow}\,\dfrac{(\mathsf{f}\,a\,\mathsf{t}) \wedge (\mathsf{t}\,a\,\mathsf{f})}{(\mathsf{f} \wedge \mathsf{t})\,a\,(\mathsf{t} \wedge \mathsf{f})}} \vee \boxed{a{\uparrow}\,\dfrac{(\mathsf{f}\,a\,\mathsf{t}) \wedge (\mathsf{t}\,a\,\mathsf{f})}{(\mathsf{f} \wedge \mathsf{t})\,a\,(\mathsf{t} \wedge \mathsf{f})}}}{\boxed{\wedge c\,\dfrac{(\mathsf{f} \wedge \mathsf{t}) \vee (\mathsf{f} \wedge \mathsf{t})}{\boxed{\dfrac{\mathsf{f} \vee \mathsf{f}}{\mathsf{f}}} \wedge \boxed{\dfrac{\mathsf{t} \vee \mathsf{t}}{\mathsf{t}}}}} \; a \; \boxed{\wedge c\,\dfrac{(\mathsf{t} \wedge \mathsf{f}) \vee (\mathsf{t} \wedge \mathsf{f})}{\boxed{\dfrac{\mathsf{t} \vee \mathsf{t}}{\mathsf{t}}} \wedge \boxed{\dfrac{\mathsf{f} \vee \mathsf{f}}{\mathsf{f}}}}}}}$$

In this reduction, we move a block of nested contractions (in purple) by creating another block of nested contractions lower in the proof and a block of nested cocontractions (in purple as well). The structure that we are therefore interested in studying is that of recursive nestings of contraction rules. For convenience and readability, we will represent these nestings in the form of a hyper-rule named *merge contraction*, which will be defined recursively in order to capture the nested structure.

Definition 7.3.2. In a ν-contractive system SA, *a merge contraction* is an SA derivation defined recursively as follows:

- A formula $A \;\nu\; B$ is a merge contraction ;
- A contraction equality rule is a merge contraction ;
- A derivation

$$c\,\dfrac{(A \;\alpha\; B) \;\nu\; (C \;\alpha\; D)}{\boxed{\begin{array}{c} A \;\nu\; C \\ \Phi_1 \Big\| \\ R \end{array}} \;\alpha\; \boxed{\begin{array}{c} B \;\nu\; D \\ \Phi_2 \Big\| \\ S \end{array}}}$$

  is a merge contraction if c is a contraction and Φ_1 and Φ_2 are merge contractions.

Definition 7.3.3. A *ν-merge* of two formulae is defined as follows:

- $A \;\nu\; B$ is a ν-merge of A and B that we call a *trivial merge*;
- u is a ν-merge of u and u, where $u \in \mathscr{U}$ is a constant;
- $C_1 \;\alpha\; C_2$ is a ν-merge of $A_1 \;\alpha\; A_2$ and $B_1 \;\alpha\; B_2$ for $\alpha \in \mathscr{R}$ if C_1 is a ν-merge of A_1 and B_1 and C_2 is a ν-merge of A_2 and B_2. In this case we say that α is the *main relation* of the merge.

If C is a ν-merge of A and B, by an abuse of language we will sometimes refer to the triple (A, B, C) as a ν-merge.

$\overline{\nu}$-merges of two formulae are defined dually.

It can be easily seen that each merge contraction corresponds to a ν-merge, and each ν-merge corresponds to a merge contraction.

Proposition 7.3.4. *Given a merge contraction* $\begin{array}{c} A \;\nu\; B \\ \phi \Big\| \\ C \end{array}$, *$C$ is a ν-merge of A and B.*

Proposition 7.3.5. *If C is a ν-merge of A and B, there is a merge contraction* $\begin{array}{c} A \;\nu\; B \\ \Big\| \\ C \end{array}$.

Exercise 7.3.6. Prove the Propositions above.

The duals of the above propositions clearly hold for $\overline{\nu}$-contractions and merge cocontractions. Given the above characterisation of merge contractions as derivations whose conclusion is a ν-merge of its premiss, for ease of notation we will represent nestings as a hyper-rule.

Definition 7.3.7. We denote merge contractions by $\mathsf{mc}{\downarrow}\,\dfrac{A\ \nu\ B}{C}$ where C is a non-trivial ν-merge of A and B.

We denote merge cocontractions by $\mathsf{mc}{\uparrow}\,\dfrac{C}{A\ \overline{\nu}\ B}$ where C is a non-trivial $\overline{\nu}$-merge of A and B.

We will permute merge contractions downwards by creating other merge contractions lower in the derivation. The main property allowing us to permute merge contractions through other rules is the given in the following proposition:

Proposition 7.3.8. *If C is a ν-merge of A and B, we can define* projections $\begin{array}{c} A \\ \pi_A \Big\| \{=,w\} \\ C \end{array}$ and $\begin{array}{c} B \\ \pi_B \Big\| \{=,w\} \\ C \end{array}$ associated to the merge.

Exercise 7.3.9. Prove the proposition above.

With the projections associated to a merge as a tool, we will now show reduction rules allowing us to permute merge (co)contractions downwards (upwards) in a proof.

Definition 7.3.10 (Reduction rule s)**.** We define the following class of reduction rules:

$$s_\rho : \quad \mathsf{mc}{\downarrow}\,\dfrac{A\ \nu\ B}{C\left\{\rho\,\dfrac{M}{N}\right\}} \quad \longrightarrow \quad \mathsf{mc}{\downarrow}\,\dfrac{\boxed{\begin{array}{c} A \\ \pi_A \Big\| \\ C\left\{\rho\,\dfrac{M}{N}\right\}\end{array}}\ \nu\ \boxed{\begin{array}{c} B \\ \pi_B \Big\| \\ C\left\{\rho\,\dfrac{M}{N}\right\}\end{array}}}{C\{N\}}$$

where π_A and π_B are the projections associated to the merge (A, B, C).

Definition 7.3.11 (Reduction rule t)**.** If the rule $\mu\,\dfrac{(A\ \nu\ B)\ \beta\ (C\ \overline{\nu}\ D)}{(A\ \beta\ C)\ \nu\ (B\ \beta\ D)}$ is derivable in SA we define the following family of rewriting rules:

$$t_\rho : \quad \rho\,\dfrac{\boxed{\mathsf{mc}{\downarrow}\,\dfrac{(A_1\ \alpha\ A_2)\ \nu\ (B_1\ \alpha\ B_2)}{C\ \alpha\ D}}\ \beta\ (E\ \alpha'\ F)}{(C\ \beta\ E)\ \alpha\ (D\ \beta'\ F)} \quad \longrightarrow \quad \mathsf{mc}{\downarrow}\,\dfrac{\mu\,\dfrac{((A_1\ \alpha\ A_2)\ \nu\ (B_1\ \alpha\ B_2))\ \beta\ \boxed{\mathsf{mc}{\uparrow}\,\dfrac{E\ \alpha'\ F}{(E\ \alpha'\ F)\ \overline{\nu}\ (E\ \alpha'\ F)}}}{\boxed{\rho\,\dfrac{(A_1\ \alpha\ A_2)\ \beta\ (E\ \alpha'\ F)}{(A_1\ \beta\ E)\ \alpha\ (A_2\ \beta'\ F)}}\ \nu\ \boxed{\rho\,\dfrac{(B_1\ \alpha\ B_2)\ \beta\ (E\ \alpha'\ F)}{(B_1\ \beta\ E)\ \alpha\ (B_2\ \beta'\ F)}}}}{(C\ \beta\ E)\ \alpha\ (D\ \beta'\ F)}$$

where C is a ν-merge of A_1 and B_1, and D is a ν-merge of A_2 and B_2.

In fact, the rewriting systems for classical logic and for multiplicative additive linear logic that allow us to permute *atomic* (co)contractions through other rules are particular instances of the generalised rewriting rules defined above.

Example 7.3.12. The reduction rule $c{\uparrow}-c{\downarrow}$ for atomic flows is an instance of reduction rule s. Likewise, the equivalent rule to permute atomic contractions and atomic cocontractions in linear logic is an instance of this reduction rule family:

$$\mathsf{mc}{\uparrow}\,\dfrac{\mathsf{mc}{\downarrow}\,\dfrac{(\bot \mathbin{a} 1)\oplus(\bot \mathbin{a} 1)}{\bot \mathbin{a} 1}}{(\bot \mathbin{a} 1)\mathbin{\&}(\bot \mathbin{a} 1)} \longrightarrow \mathsf{mc}{\downarrow}\,\dfrac{\boxed{\mathsf{mc}{\uparrow}\,\dfrac{\bot \mathbin{a} 1}{(\bot \mathbin{a} 1)\mathbin{\&}(\bot \mathbin{a} 1)}}\oplus\boxed{\mathsf{mc}{\uparrow}\,\dfrac{\bot \mathbin{a} 1}{(\bot \mathbin{a} 1)\mathbin{\&}(\bot \mathbin{a} 1)}}}{(\bot \mathbin{a} 1)\mathbin{\&}(\bot \mathbin{a} 1)}$$

or, written in terms of nestings:

$$a\bar{c}\,\dfrac{\boxed{ac\,\dfrac{((\bot\mathbin{\&}\bot)\mathbin{a}(1\mathbin{\&}1))\oplus((\bot\mathbin{\&}\bot)\mathbin{a}(1\mathbin{\&}1))}{\boxed{\mathbin{\&}c\,\dfrac{((\bot\mathbin{\&}\bot)\oplus(\bot\mathbin{\&}\bot))}{\boxed{\dfrac{\bot\oplus\bot}{\bot}}\mathbin{\&}\boxed{\dfrac{\bot\oplus\bot}{\bot}}}}\mathbin{a}\boxed{\mathbin{\&}c\,\dfrac{((1\mathbin{\&}1)\oplus(1\mathbin{\&}1))}{\boxed{\dfrac{1\oplus 1}{1}}\mathbin{\&}\boxed{\dfrac{1\oplus 1}{1}}}}}}}{(\bot \mathbin{a} 1)\mathbin{\&}(\bot \mathbin{a} 1)} \longrightarrow \mathbin{\&}c\,\dfrac{\boxed{a\bar{c}\,\dfrac{(\bot\mathbin{\&}\bot)\mathbin{a}(1\mathbin{\&}1)}{(\bot \mathbin{a} 1)\mathbin{\&}(\bot \mathbin{a} 1)}}\oplus\boxed{a\bar{c}\,\dfrac{(\bot\mathbin{\&}\bot)\mathbin{a}(1\mathbin{\&}1)}{(\bot \mathbin{a} 1)\mathbin{\&}(\bot \mathbin{a} 1)}}}{\boxed{ac\,\dfrac{(\bot \mathbin{a} 1)\oplus(\bot \mathbin{a} 1)}{\boxed{\dfrac{\bot\oplus\bot}{\bot}}\mathbin{a}\boxed{\dfrac{1\oplus 1}{1}}}}\mathbin{\&}\boxed{ac\,\dfrac{(\bot \mathbin{a} 1)\oplus(\bot \mathbin{a} 1)}{\boxed{\dfrac{\bot\oplus\bot}{\bot}}\mathbin{a}\boxed{\dfrac{1\oplus 1}{1}}}}}$$

Example 7.3.13. The reduction rule $c{\downarrow}-i{\uparrow}$ for classical logic is an instance of this reduction rule. Likewise, the equivalent reduction rule to permute atomic contractions and atomic cuts in linear logic is an instance of this reduction rule family:

$$a{\uparrow}\,\dfrac{\boxed{\mathsf{mc}{\downarrow}\,\dfrac{(\bot \mathbin{a} 1)\oplus(\bot \mathbin{a} 1)}{\bot \mathbin{a} 1}}\otimes(1\mathbin{a}\bot)}{(\bot\otimes 1)\mathbin{a}(1\otimes\bot)} \longrightarrow \mathsf{mc}{\downarrow}\,\dfrac{\oplus{\uparrow}\,\dfrac{((\bot \mathbin{a} 1)\oplus(\bot \mathbin{a} 1))\otimes\boxed{\mathsf{mc}{\uparrow}\,\dfrac{1\mathbin{a}\bot}{(1\mathbin{a}\bot)\mathbin{\&}(1\mathbin{a}\bot)}}}{\boxed{a{\uparrow}\,\dfrac{(\bot \mathbin{a} 1)\otimes(1\mathbin{a}\bot)}{(\bot\otimes 1)\mathbin{a}(1\otimes\bot)}}\oplus\boxed{a{\uparrow}\,\dfrac{(\bot \mathbin{a} 1)\otimes(1\mathbin{a}\bot)}{(\bot\otimes 1)\mathbin{a}(1\otimes\bot)}}}}{(\bot\otimes 1)\mathbin{a}(1\otimes\bot)}\ .$$

or, written in terms of nestings:

$$a{\uparrow}\,\dfrac{\boxed{ac\,\dfrac{(\bot \mathbin{a} 1)\oplus(\bot \mathbin{a} 1)}{\boxed{\dfrac{\bot\oplus\bot}{\bot}}\mathbin{a}\boxed{\dfrac{1\oplus 1}{1}}}}\otimes(1\mathbin{a}\bot)}{(\bot\otimes 1)\mathbin{a}(1\otimes\bot)} \longrightarrow ac\,\dfrac{\oplus{\uparrow}\,\dfrac{((\bot \mathbin{a} 1)\oplus(\bot \mathbin{a} 1))\otimes\boxed{a\bar{c}\,\dfrac{\boxed{\dfrac{1}{1\mathbin{\&}1}}\mathbin{a}\boxed{\dfrac{\bot}{\bot\mathbin{\&}\bot}}}{(1\mathbin{a}\bot)\mathbin{\&}(1\mathbin{a}\bot)}}}{\boxed{a{\uparrow}\,\dfrac{(\bot \mathbin{a} 1)\otimes(1\mathbin{a}\bot)}{(\bot\otimes 1)\mathbin{a}(1\otimes\bot)}}\oplus\boxed{a{\uparrow}\,\dfrac{(\bot \mathbin{a} 1)\otimes(1\mathbin{a}\bot)}{(\bot\otimes 1)\mathbin{a}(1\otimes\bot)}}}}{\boxed{\dfrac{(\bot\otimes 1)\oplus(\bot\otimes 1)}{\boxed{\dfrac{\bot\oplus\bot}{\bot}}\otimes\boxed{\dfrac{1\oplus 1}{1}}}}\mathbin{a}\boxed{\dfrac{(1\otimes\bot)\oplus(1\otimes\bot)}{\boxed{\dfrac{1\oplus 1}{1}}\otimes\boxed{\dfrac{\bot\oplus\bot}{\bot}}}}}\ .$$

We have shown that these decomposition results for classical logic and for MALL are a consequence of a wider phenomenon: both rewriting systems exploit the shape of atomic contractions to be able to permute them with other rules. Furthermore, the termination of these rewriting systems holds for the subatomic versions too: identically to the atomic versions, the subatomic rewriting systems will terminate in the absence of cycles (see Section 5). We will rigorously define and tackle cycles in the next section.

Theorem 7.3.14. *Rewriting system* C' *for* SAKS *(Figure 35) is given by the reduction rules* s *and* t *where the merge contraction being permuted has main relation* a*, and by the dual reductions.* C' *is terminating on the set of cycle-free derivations.*

Furthermore, by being able to permute generic contractions together, we advance towards proving a full decomposition theorem for classical logic and multiplicative additive linear logic, by being able to confine all contraction rules to the bottom of a proof.

Theorem 7.3.15 (Not published yet)**.** *We define rewriting system* D *for* SAKS *as the system given by the general reductions* s*,* t*, and the dual reductions for merge cocontractions. System* D *is terminating.*

7.4 Cycle elimination

Atomic contractions and atomic cocontractions can be permuted downwards/upwards in a classical logic derivation only in the absence of cycles. Cycles are created when two atom occurrences created in the same identity rule are eliminated by the same cut rule. We call these atom occurrences the *edges* of a cycle. Identically, this result holds for multiplicative additive linear logic.

Cycles are straightforwardly removed by cut-elimination. Our goal in this chapter however is to take advantage of the reductions presented in the previous chapter to show that we can remove cycles without recurring to cut-elimination, therefore proving the independence of the decomposition and the cut-elimination procedures.

Cycles can only occur due to the presence of contractions. For a cycle to occur in classical logic, two atom occurrences coming from the same introduction rule and therefore related by $\vee$ at the top of the flow have to be connected by $\wedge$ at the bottom of the flow to be eliminated by the same cut rule. Therefore, an instance of a rule that changes the relation between formulae from $\vee$ to $\wedge$ needs to occur, and it must contain the atoms involved in the cycle. The only rule that does so in subatomic system SAKS for classical logic (Figure 35) is the contraction rule m. Likewise, in multiplicative additive linear logic cycles can only occur if there is a contraction rule between the introduction and the cut of the cycle. We call these instances of contraction rules *critical.*

Definition 7.4.1. Let Φ be a derivation containing a cycle. The *critical medial* for this cycle is the lowest instance of a rule

$$\mathsf{m}\,\frac{(A\{a\} \wedge B) \vee (C \wedge D\{\bar{a}\})}{(A\{a\} \vee C) \wedge (B \vee D\{\bar{a}\})}$$

in Φ where the occurrences of a and $\bar{a}$ are the edes of the cycle.

A *critical merge contraction* is a maximal merge contraction that contains a critical medial.

The intuition behind our procedure is simple: by using the rewriting rules defined in the previous section we can permute a critical contraction rules downward until it is below the cut of its cycle. In this process derivations are significantly altered: cycles are removed and edges are bifurcated. Termination of the procedure is easy to check: we show that when permuting critical contractions downwards we do not create any additional critical contractions.

Theorem 7.4.2. *Let Φ be a derivation with n critical merge contractions. Then there exists a derivation Ψ with the same premiss and conclusion with $n-1$ critical merge contractions.*

To eliminate all cycles from a derivation, one simply performs the procedure n times, once for each critical merge contraction.

Corollary 7.4.3. *Given a derivation Φ, there exists a derivation Ψ with the same premiss and conclusion and without cycles.*

7.5 Notes

The subatomic methodology is base on an idea by Alessio Guglielmi [Gug02]. Its formal definition, and the splitting, decomposition and cycle-elimination results presented here are taken from Andrea Aler Tubella's thesis [Ale16]. A complete account of subatomic splitting by Guglielmi and Aler Tubella can be found in [AG17].

Merge contractions have a lot of interesting and unusual properties, which are expanded upon in Ben Ralph's thesis in [Ral19], where another cycle-elimination procedure is shown as well. Another procedure based on the same idea as the one we presented in these notes is presented by Aler Tubella, Guglielmi and Ralph in [AGR17]. In fact, the phenomenon of cycles has also been studied in the sequent calculus, where it has been shown that it is possible to remove them through a procedure of quadratic-time complexity [Car02].

Subatomic logic is a recent advance, and the focus of ongoing research. Some recent work by Guglelmi and Chris Barrett on applying subatomic methods to describe and operate with decision trees can be found in [BG19]. New work by Luca Roversi on extending subatomic logic in order to study systems with different SAT-complexities can be found in [Rov18].

8 Final Remarks

There are many exciting aspects of deep inference that have not been touched in this short course. With only 5x90min lecture time, some selection had to be made. In this final section, we will give a list of some topics that have been left out, and we also give references for the interested student to get more information on these topics.

- First, it should be mentioned that deep inference systems exist not only for classical and linear logic, but also for intuitionistic logic [Tiu06a, GS14], modal logics [SS05, Sto07], hybrid logic [Str07b], the logic of bunched implications [Hor06], and first-order logic [GS14, Ral18, Ral19]. Ralph's PhD-thesis [Ral19] also investigates the relation between deep inference decomposition theorems and Herbrand's theorem.

- Another important aspect of deep inference is *proof complexity*, which studies the size of proofs and compares proof systems with respect to the size of the proofs they produce. Due to the flexiblity in the design of inference rules, deep inference systems can outperform other formalisms with respect to the size of the produced proofs. This has first been observed in [BG09]. Follow-up results on the proof complexity of deep inference can be found in [Str12, NS15, Das11, Das12, Das15].

- We can also see (linear) inference rules of SKS as rewrite rules of a linear rewrite system and study its properties. This has been done in [Das13], and has led to interesting complexity results [DS15, DS16, BS17].

- Insights from deep inference can also help to study category theoretical aspects of classical proof theory [Str07c, Lam07].

- An important recent development is the *atomic λ-calculus* [GHP13b, GHP13a] that is in Curry-Howard correspondence with a deep inference proof system for intuitionistic logic. This calculus has been extended to a typeable calculus with explicit sharing which extends the Curry-Howard interpretation of open deduction [She19]. Similarly, the atomic *atomic $\lambda\mu$-calculus* [He18] offers a Curry-Howard interpretation of a classical deep inference proof system.

- Finally, deep inference proof systems can be used to describe process calculi, in such a way that the proof of a formula corresponds to the trace of a process in the so-called *proof as process* paradigm [Bru02, HTAC16, HT19, HTAC19].

An up-to-date list of references can also be found at the deep inference webpage

`http://alessio.guglielmi.name/res/cos/index.html`

maintained by Alessio Guglielmi.

9 References

[AG17] Andrea Aler Tubella and Alessio Guglielmi. Subatomic Proof Systems: Splittable Systems. *ACM Transactions on Computational Logic*, 19(1:5):1–33, mar 2017.

[AGR17] Andrea Aler Tubella, Alessio Guglielmi, and Benjamin Ralph. Removing Cycles from Proofs. In *26th EACSL Annual Conference on Computer Science Logic (CSL 2017)*, volume 82, sep 2017.

[Ale16] Andrea Aler Tubella. *A study of normalisation through subatomic logic*. Thesis, University of Bath, dec 2016.

[BG09] Paola Bruscoli and Alessio Guglielmi. On the proof complexity of deep inference. *ACM Transactions on Computational Logic*, 10(2):1–34, 2009. Article 14.

[BG19] C Barrett and A Guglielmi. A Subatomic Proof System for Decision Trees. Technical report, University of Bath, 2019.

[Bru02] Paola Bruscoli. A purely logical account of sequentiality in proof search. In Peter J. Stuckey, editor, *Logic Programming, 18th International Conference*, volume 2401 of *Lecture Notes in Artificial Intelligence*, pages 302–316. Springer-Verlag, 2002.

[Brü03] Kai Brünnler. *Deep Inference and Symmetry for Classical Proofs*. PhD thesis, Technische Universität Dresden, 2003.

[Brü06] Kai Brünnler. Locality for Classical Logic. *Notre Dame Journal of Formal Logic*, 47(4):557–580, 2006.

[BS17] Paola Bruscoli and Lutz Straßburger. On the length of medial-switch-mix derivations. In Juliette Kennedy and Ruy J. G. B. de Queiroz, editors, *Logic, Language, Information, and Computation - 24th International Workshop, Proceedings*, volume 10388 of *Lecture Notes in Computer Science*, pages 68–79. Springer, 2017.

[BT01] Kai Brünnler and Alwen Fernanto Tiu. A local system for classical logic. In R. Nieuwenhuis and A. Voronkov, editors, *LPAR 2001*, volume 2250 of *LNAI*, pages 347–361. Springer, 2001.

[BvdW95] Gianluigi Bellin and Jacques van de Wiele. Subnets of proof-nets in MLL^-. In J.-Y. Girard, Y. Lafont, and L. Regnier, editors, *Advances in Linear Logic*, volume 222 of *London Mathematical Society Lecture Notes*, pages 249–270. Cambridge University Press, 1995.

[Car02] A. Carbone. The cost of a cycle is a square. *Journal of Symbolic Logic*, 67(1):35–60, mar 2002.

[CGS11] Kaustuv Chaudhuri, Nicolas Guenot, and Lutz Straßburger. The Focused Calculus of Structures. In Marc Bezem, editor, *Computer Science Logic (CSL)*, volume 12 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 159–173. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2011.

[Das11] Anupam Das. On the proof complexity of cut-free bounded deep inference. In Kai Brünnler and George Metcalfe, editors, *Automated Reasoning with Analytic Tableaux and Related Methods - 20th International Conference, TABLEAUX 2011*, volume 6793 of *LNCS*, pages 134–148. Springer, 2011.

[Das12] Anupam Das. Complexity of deep inference via atomic flows. In S. Barry Cooper, Anuj Dawar, and Benedikt Löwe, editors, *Computability in Europe*, volume 7318 of *LNCS*, pages 139–150. Springer-Verlag, 2012.

[Das13] Anupam Das. Rewriting with linear inferences in propositional logic. In Femke van Raamsdonk, editor, *24th International Conference on Rewriting Techniques and Applications (RTA)*, volume 21 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 158–173. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2013.

[Das15] Anupam Das. On the relative proof complexity of deep inference via atomic flows. *Logical Methods in Computer Science*, 11(1), 2015.

[DR89] Vincent Danos and Laurent Regnier. The structure of multiplicatives. *Arch. Math. Log.*, 28(3):181–203, 1989.

[DS15] Anupam Das and Lutz Straßburger. No complete linear term rewriting system for propositional logic. In Maribel Fernández, editor, *26th International Conference on Rewriting Techniques and Applications, RTA 2015, June 29 to July 1, 2015, Warsaw, Poland*, volume 36 of *LIPIcs*, pages 127–142, 2015.

[DS16] Anupam Das and Lutz Straßburger. On linear rewriting systems for boolean logic and some applications to proof theory. *Logical Methods in Computer Science*, 12(4), 2016.

[Duf65] R.J Duffin. Topology of series-parallel networks. *Journal of Mathematical Analysis and Applications*, 10(2):303 – 318, 1965.

[Gen35a] Gerhard Gentzen. Untersuchungen über das logische Schließen. I. *Mathematische Zeitschrift*, 39:176–210, 1935.

[Gen35b] Gerhard Gentzen. Untersuchungen über das logische Schließen. II. *Mathematische Zeitschrift*, 39:405–431, 1935.

[GG08] Alessio Guglielmi and Tom Gundersen. Normalisation control in deep inference via atomic flows. *Logical Methods in Computer Science*, 4(1:9):1–36, 2008.

[GGP10] Alessio Guglielmi, Tom Gundersen, and Michel Parigot. A Proof Calculus Which Reduces Syntactic Bureaucracy. In Christopher Lynch, editor, *21st International Conference on Rewriting Techniques and Applications (RTA)*, volume 6 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 135–150. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2010.

[GGS10] Alessio Guglielmi, Tom Gundersen, and Lutz Straßburger. Breaking paths in atomic flows for classical logic. In *Proceedings of the 25th Annual IEEE Symposium on Logic in Computer Science, LICS 2010, 11-14 July 2010, Edinburgh, United Kingdom*, pages 284–293. IEEE Computer Society, 2010.

[GHP13a] Tom Gundersen, Willem Heijltjes, and Michel Parigot. A Proof of Strong Normalisation of the Typed Atomic Lambda-Calculus. In Ken McMillan, Aart Middeldorp, and Andrei Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning (LPAR-19)*, volume 8312 of *Lecture Notes in Computer Science*, pages 340–354. Springer-Verlag, 2013.

[GHP13b] Tom Gundersen, Willem Heijltjes, and Michel Parigot. Atomic Lambda Calculus: {A} Typed Lambda-Calculus with Explicit Sharing. In Orna Kupferman, editor, *28th Annual IEEE Symposium on Logic in Computer Science (LICS)*, pages 311–320. IEEE, 2013.

[Gir87] Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

[Gir95] Jean-Yves Girard. Linear logic: its syntax and semantics. In Jean-Yves Girard, Yves Lafont, and Laurent Regnier, editors, *Advances in Linear Logic*, pages 1–42. Cambridge University Press, 1995.

[Göd31] Kurt Gödel. Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. *Monatshefte für Mathematik und Physik*, 38:173–198, 1931.

[GS01] Alessio Guglielmi and Lutz Straßburger. Non-commutativity and MELL in the calculus of structures. In Laurent Fribourg, editor, *Computer Science Logic, CSL 2001*, volume 2142 of *LNCS*, pages 54–68. Springer-Verlag, 2001.

[GS02] Alessio Guglielmi and Lutz Straßburger. A non-commutative extension of MELL. In Matthias Baaz and Andrei Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning, LPAR 2002*, volume 2514 of *LNAI*, pages 231–246. Springer-Verlag, 2002.

[GS11] Alessio Guglielmi and Lutz Straßburger. A system of interaction and structure V: the exponentials and splitting. *Mathematical Structures in Computer Science*, 21(3):563–584, 2011.

[GS14] Nicolas Guenot and Lutz Straßburger. Symmetric Normalisation for Intuitionistic Logic. In Thomas Henzinger and Dale Miller, editors, *Joint Meeting of the 23rd EACSL Annual Conference on Computer Science Logic (CSL) and the 29th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 45:1—-10. ACM, 2014.

[Gug02] Alessio Guglielmi. Subatomic Logic, 2002.

[Gug07] Alessio Guglielmi. A system of interaction and structure. *ACM Transactions on Computational Logic*, 8(1):1–64, 2007.

[Gun09] Tom Gundersen. *A General View of Normalisation through Atomic Flows.* PhD thesis, The University of Bath, 2009.

[HA28] David Hilbert and Wilhelm Ackermann. *Grundzüge der theoretischen Logik*, volume XXVII of *Die Grundlehren der Mathematischen Wissenschaften.* Verlag von Julius Springer, 1928.

[He18] Fanny He. *The Atomic Lambda-Mu Calculus.* PhD thesis, University of Bath, 2018.

[Hil22] David Hilbert. Die logischen Grundlagen der Mathematik. *Mathematische Annalen*, 88:151–165, 1922.

[Hor06] Benjamin Robert Horsfall. *The Logic of Bunched Implications: {A} Memoir.* PhD thesis, University of Melbourne, 2006.

[Hor19] Ross Horne. The Sub-Additives: A Proof Theory for Probabilistic Choice extending Linear Logic. In *FSCD*, 2019.

[HT19] Ross Horne and Alwen Tiu. Constructing weak simulations from linear implications for processes with private names. *Mathematical Structures in Computer Science*, 29(8):1275–1308, 2019.

[HTAC16] Ross Horne, Alwen Tiu, Bogdan Aman, and Gabriel Ciobanu. Private Names in Non-Commutative Logic. In Josée Desharnais and Radha Jagadeesan, editors, *27th International Conference on Concurrency Theory (CONCUR)*, volume 59 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 31:1–31:16. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2016.

[HTAC19] Ross Horne, Alwen Tiu, Bogdan Aman, and Gabriel Ciobanu. De Morgan Dual Nominal Quantifiers Modelling Private Names in Non-Commutative Logic. *ACM Trans. Comput. Logic*, 20(4):22:1–22:44, July 2019.

[Hug06a] Dominic Hughes. Proofs Without Syntax. *Annals of Mathematics*, 164(3):1065–1076, 2006.

[Hug06b] Dominic Hughes. Towards Hilbert's 24^{th} problem: Combinatorial proof invariants: (preliminary version). *Electr. Notes Theor. Comput. Sci.*, 165:37–63, 2006.

[Laf95] Yves Lafont. *Equational reasoning with 2-dimensional diagrams*, volume 909 of *LNCS*, pages 170–195. 1995.

[Lam07] François Lamarche. Exploring the gap between linear and classical logic. *Theory and Applications of Categories*, 18(18):473–535, 2007.

[Möh89] Rolf H. Möhring. Computationally tractable classes of ordered sets. In I. Rival, editor, *Algorithms and Order*, pages 105–194. Kluwer Acad. Publ., 1989.

[NS15] Novak Novakovic and Lutz Straßburger. On the power of substitution in the calculus of structures. *ACM Trans. Comput. Log.*, 16(3):19, 2015.

[Pra65] Dag Prawitz. *Natural Deduction, A Proof-Theoretical Study.* Almquist and Wiksell, 1965.

[Ral18] Benjamin Ralph. A natural proof system for herbrand's theorem. In Sergei N. Artëmov and Anil Nerode, editors, *Logical Foundations of Computer Science - International Symposium, LFCS 2018, Proceedings*, volume 10703 of *Lecture Notes in Computer Science*, pages 289–308. Springer, 2018.

[Ral19] Benjamin Ralph. *Modular Normalisation of Classical Proofs.* PhD thesis, University of Bath, 2019.

[Ret93] Christian Retoré. *Réseaux et Séquents Ordonnés.* PhD thesis, Université Paris VII, 1993.

[Ret03] Christian Retoré. Handsome proof-nets: perfect matchings and cographs. *Theoretical Computer Science*, 294(3):473–488, 2003.

[Rov18] Luca Roversi. Subatomic systems need not be subatomic. Preprint, 2018.

[SG11] Lutz Straßburger and Alessio Guglielmi. A system of interaction and structure IV: The exponentials and decomposition. *ACM Trans. Comput. Log.*, 12(4):23, 2011.

[She19] David Sherratt. *A lambda-calculus that achieves full laziness with spine duplication.* PhD thesis, University of Bath, 2019.

[SS05] Charles Stewart and Phiniki Stouppa. A systematic proof theory for several modal logics. In R. A. Schmidt, I. Pratt-Hartmann, M. Reynolds, and H. Wansing, editors, *Advances in Modal Logic, Volume 5*, pages 309–333. King's College Publications, 2005.

[Sto07] Phiniki Stouppa. A deep inference system for the modal logic S5. *Studia Logica*, 85(2):199–214, 2007.

[Str02a] Lutz Straßburger. A local system for linear logic. In Matthias Baaz and Andrei Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning, LPAR 2002*, volume 2514 of *LNAI*, pages 388–402. Springer-Verlag, 2002.

[Str02b] Lutz Straßburger. A Local System for Linear Logic. In Matthias Baaz and Andrei Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning (LPAR)*, volume 2514 of *Lecture Notes in Computer Science*, pages 388–402. Springer-Verlag, 2002.

[Str03a] Lutz Straßburger. *Linear Logic and Noncommutativity in the Calculus of Structures.* PhD thesis, Technische Universität Dresden, 2003.

[Str03b] Lutz Straßburger. MELL in the Calculus of Structures. *Theoretical Computer Science*, 309(1–3):213–285, 2003.

[Str07a] Lutz Straßburger. A characterisation of medial as rewriting rule. In Franz Baader, editor, *Term Rewriting and Applications, RTA'07*, volume 4533 of *LNCS*, pages 344–358. Springer, 2007.

[Str07b] Lutz Straßburger. Deep inference for hybrid logic. In *International Workshop on Hybrid Logic 2007 (Part of ESSLLI'07)*, 2007.

[Str07c] Lutz Straßburger. On the axiomatisation of Boolean categories with and without medial. *Theory and Applications of Categories*, 18(18):536–601, 2007.

[Str12] Lutz Straßburger. Extension without cut. *Annals of Pure and Applied Logic*, 163(12):1995–2007, 2012.

[Str17] Lutz Straßburger. Combinatorial flows and their normalisation. In Dale Miller, editor, *2nd International Conference on Formal Structures for Computation and Deduction, FSCD 2017, September 3-9, 2017, Oxford, UK*, volume 84 of *LIPIcs*, pages 31:1–31:17. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

[Tiu06a] Alwen Tiu. A local system for intuitionistic logic. In M. Hermann and A. Voronkov, editors, *LPAR 2006*, volume 4246 of *Lecture Notes in Artificial Intelligence*, pages 242–256. Springer-Verlag, 2006.

[Tiu06b] Alwen Tiu. A system of interaction and structure II: The Need for Deep Inference. *Logical Methods in Computer Science*, 2(2):4:1—-24, 2006.

[TS00] Anne Sjerp Troelstra and Helmut Schwichtenberg. *Basic Proof Theory.* Cambridge University Press, second edition, 2000.